```
MMM         MMM     000000000     MMM         MMM
MMM         MMM     000000000     MMM         MMM
MMM         MMM     000000000     MMM         MMM
MMMMMM   MMMMMM  000         000  MMMMMM   MMMMMM
MMMMMM   MMMMMM  000         000  MMMMMM   MMMMMM
MMMMMM   MMMMMM  000         000  MMMMMM   MMMMMM
MMM   MMM   MMM  000         000  MMM   MMM   MMM
MMM   MMM   MMM  000         000  MMM   MMM   MMM
MMM   MMM   MMM  000         000  MMM   MMM   MMM
MMM         MMM  000         000  MMM         MMM
MMM         MMM  000         000  MMM         MMM
MMM         MMM  000         000  MMM         MMM
MMM         MMM  000         000  MMM         MMM
MMM         MMM  000         000  MMM         MMM
MMM         MMM  000         000  MMM         MMM
MMM         MMM  000         000  MMM         MMM
MMM         MMM  000         000  MMM         MMM
MMM         MMM  000         000  MMM         MMM
MMM         MMM     000000000     MMM         MMM
MMM         MMM     000000000     MMM         MMM
MMM         MMM     000000000     MMM         MMM
```

BCDEFGHIJKLMNBCDEFGHIJKLMNBCDEFGHIJKLMNBCDEFGHIJKLMNBCDEFGHI

```
**FILE**ID**MOMSERVIC                                    G 10



MM      MM    000000    MM      MM    SSSSSSSS  EEEEEEEEEE  RRRRRRRR    VV      VV    IIIIII      CCCCCCCC
MM      MM    000000    MM      MM    SSSSSSSS  EEEEEEEEEE  RRRRRRRR    VV      VV    IIIIII      CCCCCCCC
MMMM  MMMM  00      00  MMMM  MMMM  SS          EE          RR      RR  VV      VV      II      CC
MMMM  MMMM  00      00  MMMM  MMMM  SS          EE          RR      RR  VV      VV      II      CC
MM  MM  MM  00      00  MM  MM  MM  SS          EE          RR      RR  VV      VV      II      CC
MM  MM  MM  00      00  MM  MM  MM  SS          EE          RR      RR  VV      VV      II      CC
MM      MM  00      00  MM      MM    SSSSSS    EEEEEEEE    RRRRRRRR    VV      VV      II      CC
MM      MM  00      00  MM      MM    SSSSSS    EEEEEEEE    RRRRRRRR    VV      VV      II      CC
MM      MM  00      00  MM      MM          SS  EE          RR  RR      VV      VV      II      CC
MM      MM  00      00  MM      MM          SS  EE          RP  RR      VV      VV      II      CC
MM      MM  00      00  MM      MM          SS  EE          RR    RR      VV  VV        II      CC             ....
MM      MM  00      00  MM      MM          SS  EE          RR    RR      VV  VV        II      CC             ....
MM      MM    000000    MM      MM  SSSSSSSS    EEEEEEEEEE  RR      RR      VV        IIIIII      CCCCCCCC     ....
MM      MM    000000    MM      MM  SSSSSSSS    EEEEEEEEEE  RR      RR      VV        IIIIII      CCCCCCCC     ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 %TITLE 'Network Management Maintenance Operations Service routines'
0 MODULE MOMSERVICE (
0                       LANGUAGE (BLISS32),
0                       ADDRESSING_MODE (NONEXTERNAL=LONG_RELATIVE),
0                       ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE),
0                       IDENT = 'V04-000'
0                       ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:  DECnet-VAX V2.0 Maintenance Operations Module
1 !
1 ! ABSTRACT:
1 !       This module contains routines to handle maintenance operations
1 !       such as trigger, dump, and line and circuit loop.
1 !
1 ! ENVIRONMENT:  VAX/VMS Operating System
1 !
1 ! AUTHOR:  Kathy Perko
1 !
1 ! CREATION DATE:  5-Jan-1983
1 !
1 ! MODIFIED BY:
1 !       V03-004 MKP0004         Kathy Perko             12-June-1984
1 !               Add the Ethernet address to logging events.
1 !
1 !       V03-003 MKP0003         Kathy Perko             28-April-1984
1 !               Transmit a "dump complete" message at the end of an upline
1 !               dump.  Also, for triggers, if the target does not respond
1 !               to a system ID message, send the trigger message to both
1 !               the hardware NI address and the DECnet NI address.
1 !
1 !       V03-002 MKP0002         Kathy Perko             11-Feb-1984
```

```
;    58        0058  1 !                  Use Remote console protocol on Ethernet for boot messages
;    59        0059  1 !                  instead of load/dump protocol.  Also, fix parsing of
;    60        0060  1 !                  autoservice initial MOP message to check for loop assistance
;    61        0061  1 !                  requests.
;    62        0062  1 !
;    63        0063  1 !          V03-001 MKP0001         Kathy Perko             8-May-1983
;    64        0064  1 !                  Add node id to event logged for Aborted Service Request.
;    65        0065  1 !                  Fix the trigger at the beginning of LOAD to try both
;    66        0066  1 !                  the HIORD and the hardware address of the target.
;    67        0067  1 !
;    68        0068  1 !--
;    69        0069  1
;    71         0070  1 %SBTTL 'Declarations'
;    72         0071  1
;    73         0072  1 !
;    74         0073  1 ! TABLE OF CONTENTS:
;    75         0074  1 !
;    76         0075  1
;    77         0076  1 FORWARD ROUTINE
;    78         0077  1     mom$operservice     : NOVALUE,
;    79         0078  1     mom$autoservice     : NOVALUE,
;    80         0079  1     mom_get_initial_mop_msg: NOVALUE,
;    81         0080  1     mom_volunteer_assistance: NOVALUE,
;    82         0081  1     mom$dump            : NOVALUE,
;    83         0082  1     mom$trigger,
;    84         0083  1     mom$servicehandler,
;    85         0084  1     mom$autohandler,
;    86         0085  1     mom$log_event       : NOVALUE,
;    87         0086  1     mom$chk_mop_error   : NOVALUE;
;    88         0087  1
;    89         0088  1 !
;    90         0089  1 ! INCLUDE FILES:
;    91         0090  1 !
;    92         0091  1
;    93         0092  1 LIBRARY 'LIB$:MOMLIB.L32';
;    94         0093  1 LIBRARY 'SHRLIB$:NMALIBRY.L32';
;    95         0094  1 LIBRARY 'SHRLIB$:EVCDEF.L32';
;    96         0095  1 LIBRARY 'SHRLIB$:NET.L32';
;    97         0096  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    98         0097  1
;    99         0098  1 !
;   100         0099  1 ! EXTERNAL REFERENCES:
;   101         0100  1 !
;   102         0101  1
;   103         0102  1 $mom_externals;                         ! Macro to define common externals
;   104         0103  1
;   105         0104  1 EXTERNAL LITERAL
;   106         0105  1     mom$_unsmopdev,
;   107         0106  1     mom$_alpbfovf,
;   108         0107  1     mdt$gk_mopdevcnt;
;   109         0108  1
;   110         0109  1 EXTERNAL
;   111         0110  1     mom$ab_mopdevices : BBLOCKVECTOR [0,mdt$k_entrylen],
;   112         0111  1     mom$npa_init,
;   113         0112  1     mom$npa_load,
;   114         0113  1     mom$npa_mopinit,
;   115         0114  1     mom$npa_mopdump,
;   116         0115  1     mom$npa_trigger;
;   117         0116  1
;   118         0117  1 EXTERNAL ROUTINE
;   119         0118  1     nma$nparse,
;   120         0119  1     mom$parse_nice_entity,
;   121         0120  1     mom$test,
;   122         0121  1     mom$load,
;   123         0122  1     mom$bld_reply,
;   124         0123  1     mom$bldmopboot,
;   125         0124  1     mom$bldmoprds,
;   126         0125  1     mom$bldmopplt,
;   127         0126  1     mom$debug_msg,
;  128            0127   1        mom$debug_txt,
;  129            0128   1        mom$error,
;  130            0129   1        mom$getsrvdata,
;  131            0130   1        mom$getsrvtimer,
;  132            0131   1        mom$loophandler,
;  133            0132   1        mom$init_CIB,
;  134            0133   1        mom$mopsetsubstate,
;  135            0134   1        mom$mopopen,
;  136            0135   1        mom$mopsend,
;  137            0136   1        mom$mop_receive_qiow,
;  138            0137   1        mom$mopsndrcv,
;  139            0138   1        mom$passiveloop,
;  140            0139   1        mom$netacp_qio,
;  141            0140   1        mom$send,
;  142            0141   1        mom$set_NI_addr,
;  143            0142   1        mom$srvopen,
;  144            0143   1        mom$srvwrite,
;  145            0144   1        mom$srvrewind,
;  146            0145   1        mom$srvclose;
;  147            0146   1
;  148            0147   1
;  150          0148  1 %SBTTL 'mom$operservice      Parse the maintenance request (NICE) message'
;  151          0149  1 GLOBAL ROUTINE mom$operservice : NOVALUE =
;  152          0150  1
;  153          0151  1 !++
;  154          0152  1 ! FUNCTIONAL DESCRIPTION:
;  155          0153  1 !
;  156          0154  1 !       This routine parses the NICE service request message and
;  157          0155  1 !       builds the permanent data base record to hold the data.
;  158          0156  1 !
;  159          0157  1 ! IMPLICIT OUTPUTS:
;  160          0158  1 !
;  161          0159  1 !       MOM$GQ_SRVDATDSC describes the service parameter information from the
;  162          0160  1 !               volatile data base and from the NICE message.
;  163          0161  1 !
;  164          0162  1 ! ROUTINE VALUE:
;  165          0163  1 ! COMPLETION CODES:
;  166          0164  1 !
;  167          0165  1 !       Signal errors.
;  168          0166  1 !
;  169          0167  1 ! SIDE EFFECTS:
;  170          0168  1 !
;  171          0169  1 !       The service circuit is open for MOP operations.
;  172          0170  1 !
;  173          0171  1 !--
;  174          0172  1
;  175          0173  2 BEGIN
;  176          0174  2
;  177          0175  2 LOCAL
;  178          0176  2     mom_a_npatbl,                           ! Pointer to NPARSE table
;  179          0177  2     mom_a_routine,                          ! Address of service routine
;  180          0178  2     mom_l_msgsize,                          ! Size of response message
;  181          0179  2     mom_l_status;                           ! General purpose status
;  182          0180  2
;  183          0181  2 !
;  184          0182  2 ! Enable condition handler for cleanup.
;  185          0183  2 !
;  186          0184  2 ENABLE mom$servicehandler;
;  187          0185  2 !
;  188          0186  2 ! Parse the NICE command message up the parameters.  This parsing includes
;  189          0187  2 ! the function code, the option byte, and the target entity ID.
;  190          0188  2 !
;  191          0189  2 mom$parse_nice_entity ();
;  192          0190  2 !
;  193          0191  2 ! Select parse table according to function code.
;  194          0192  2 !
;  195          0193  2 SELECTONEU .mom$gb_function OF
;  196          0194  2     SET
;  197          0195  2     [nma$c_fnc_loa]:            ! Load
;  198          0196  3         BEGIN
;  199          0197  3         mom_a_routine = mom$load;
;  200          0198  3         mom_a_npatbl = mom$npa_load;
;  201          0199  3         END;
;  202          0200  2
;  203          0201  2     [nma$c_fnc_tri]:            ! Trigger
;  204          0202  3         BEGIN
;  205          0203  3         mom_a_routine = mom$trigger;
;  206          0204  3         mom_a_npatbl = mom$npa_trigger;
;  207          0205  2             END;
;  208          0206  2
;  209          0207  2         [nma$c_fnc_tes]:                ! Loop circuit, node, or line
;  210          0208  3             BEGIN
;  211          0209  3             mom$test ();
;  212          0210  3             RETURN;
;  213          0211  2             END;
;  214          0212  2
;  215          0213  2         [OTHERWISE]:                            ! Invalid function
;  216          0214  3             BEGIN
;  217          0215  3             mom$error (nma$c_sts_fun);
;  218          0216  2             END;
;  219          0217  2
;  220          0218  2         TES;
;  221          0219  2     !
;  222          0220  2     ! Get the node's service parameters from the volatile data base and
;  223          0221  2     ! put them into the Service Data Table.  Then get the parameters
;  224          0222  2     ! (if any) from the NICE command message and overwrite the appropriate
;  225          0223  2     ! Service Data Table.  This is done because the NICE message parameters
;  226          0224  2     ! take precedence over the ones in the volatile database.
;  227          0225  2     !
;  228          0226  2     nma$nparse (mom$ab_nparse_blk, .mom_a_npatbl);
;  229          0227  2     !
;  230          0228  2     ! Set up the event in case of an aborted service request.
;  231          0229  2     !
;  232          0230  2     mom$gw_evt_code = evc$c_nma_abs;        ! Event code (aborted service request)
;  233          0231  2     mom$gb_evt_prsn = evc$c_nma_prsn_loe;   ! Circuit open error
;  234          0232  2
;  235          0233  2     mom$getsrvdata ();! Build data base from volatile
;  236          0234  2     !
;  237          0235  2     ! Get the service line timer (used to time all interactions with the
;  238          0236  2     ! target node).
;  239          0237  2     !
;  240          0238  2     mom$getsrvtimer ();
;  241          0239  2     !
;  242          0240  2     ! Dispatch the function to the service routine.
;  243          0241  2     ! (The service routine will signal the status and cleanup will
;  244          0242  2     ! be performed by the condition handler.)
;  245          0243  2     !
;  246          0244  2     (.mom_a_routine) (mom$ab_cib);
;  247          0245  2
;  248          0246  1     END;                            ! End of MOM$OPERSERVICE


                                                        .TITLE  MOMSERVICE Network Management Maintenance Opera
;                                                                          tions Servi
                                                        .IDENT  \V04-000\

                                                        .EXTRN  MOM$GL_LOGMASK, MOM$GL_SVD_INDEX
                                                        .EXTRN  MOM$AB_SERVICE_DATA
                                                        .EXTRN  MOM$GB_FUNCTION
                                                        .EXTRN  MOM$GB_OPTION_BYTE
                                                        .EXTRN  MOM$GB_ENTITY_CODE
                                                        .EXTRN  MOM$AB_ENTITY_BUF
                                                        .EXTRN  MOM$GQ_ENTITY_BUF_DSC
                                                        .EXTRN  MOM$GL_SERVICE_FLAGS
                                                        .EXTRN  MOM$AB_NPARSE_BLK
```

```
.EXTRN  MOM$AB_NICE_RCV_BUF
.EXTRN  MOM$AB_NICE_XMIT_BUF
.EXTRN  MOM$GQ_NICE_RCV_BUF_DSC
.EXTRN  MOM$GL_NICE_RCV_MSG_LEN
.EXTRN  MOM$GQ_NICE_XMIT_BUF_DSC
.EXTRN  MOM$AB_MSGBLOCK
.EXTRN  MOM$AB_ACPQIO_BUFFER
.EXTRN  MOM$GQ_ACPQIO_BUF_DSC
.EXTRN  MOM$AB_CIB, MOM$AB_LOOP_CIB
.EXTRN  MOM$AB_TRIGGER_CIB
.EXTRN  MOM$AB_MOP_XMIT_BUF
.EXTRN  MOM$GQ_MOP_XMIT_BUF_DSC
.EXTRN  MOM$AB_MOP_RCV_BUF
.EXTRN  MOM$GQ_MOP_RCV_BUF_DSC
.EXTRN  MOM$AB_MOP_MSG, MOM$GQ_MOP_MSG_DSC
.EXTRN  MOM$GW_EVT_CODE
.EXTRN  MOM$GB_EVT_POPR
.EXTRN  MOM$GB_EVT_PRSN
.EXTRN  MOM$GB_EVT_PSER
.EXTRN  SVD$GK_PCNO_ADD
.EXTRN  SVD$GK_PCNO_SDV
.EXTRN  SVD$GK_PCNO_CPU
.EXTRN  SVD$GK_PCNO_STY
.EXTRN  SVD$GK_PCNO_DAD
.EXTRN  SVD$GK_PCNO_DCT
.EXTRN  SVD$GK_PCNO_IHO
.EXTRN  SVD$GK_PCNO_NNA
.EXTRN  SVD$GK_PCNO_SLI
.EXTRN  SVD$GK_PCNO_SPA
.EXTRN  SVD$GK_PCNO_HWA
.EXTRN  SVD$GK_PCNO_SNV
.EXTRN  SVD$GK_PCNO_LOA
.EXTRN  SVD$GK_PCNO_SLO
.EXTRN  SVD$GK_PCNO_TLO
.EXTRN  SVD$GK_PCNO_DFL
.EXTRN  SVD$GK_PCNO_SID
.EXTRN  SVD$GK_PCNO_DUM
.EXTRN  SVD$GK_PCNO_SDU
.EXTRN  SVD$GK_PCNO_$HNA
.EXTRN  SVD$GK_PCNO_$HHW
.EXTRN  SVD$GK_PCNO_$FTY
.EXTRN  SVD$GK_PCNO_PHA
.EXTRN  SVD$GK_PCNO_$DA
.EXTRN  SVD$GK_PCNO_LPC
.EXTRN  SVD$GK_PCNO_LPL
.EXTRN  SVD$GK_PCNO_LPD
.EXTRN  SVD$GK_PCNO_LPH
.EXTRN  SVD$GK_PCNO_LPA
.EXTRN  SVD$GK_PCNO_LPN
.EXTRN  SVD$GK_PCNO_$LNA
.EXTRN  SVD$GK_PCNO_$LNH
.EXTRN  SVD$GK_PCNO_LAN
.EXTRN  SVD$GK_PCNO_$LNN
.EXTRN  SVD$GK_PCNO_$LAH
.EXTRN  SVD$GK_PCLI_STI
.EXTRN  SVD$C_ENTRY_COUNT
.EXTRN  MOM$_UNSMOPDEV, MOM$_ALPBFOVF
```

```
                                                           .EXTRN   MDT$GK_MOPDEVCNT
                                                           .EXTRN   MOM$AB_MOPDEVICES
                                                           .EXTRN   MOM$NPA_INIT, MOM$NPA_LOAD
                                                           .EXTRN   MOM$NPA_MOPINIT
                                                           .EXTRN   MOM$NPA_MOPDUMP
                                                           .EXTRN   MOM$NPA_TRIGGER
                                                           .EXTRN   NMA$NPARSE, MOM$PARSE_NICE_ENTITY
                                                           .EXTRN   MOM$TEST, MOM$LOAD
                                                           .EXTRN   MOM$BLD_REPLY, MOM$BLDMOPBOOT
                                                           .EXTRN   MOM$BLDMOPRDS, MOM$BLDMOPPLT
                                                           .EXTRN   MOM$DEBUG_MSG, MOM$DEBUG_TXT
                                                           .EXTRN   MOM$ERROR, MOM$GETSRVDATA
                                                           .EXTRN   MOM$GETSRVTIMER
                                                           .EXTRN   MOM$LOOPHANDLER
                                                           .EXTRN   MOM$INIT_CIB, MOM$MOPSETSUBSTATE
                                                           .EXTRN   MOM$MOPOPEN, MOM$MOPSEND
                                                           .EXTRN   MOM$MOP_RECEIVE_QIOW
                                                           .EXTRN   MOM$MOPSNDRCV, MOM$PASSIVELOOP
                                                           .EXTRN   MOM$NETACP_QIO, MOM$SEND
                                                           .EXTRN   MOM$SET_NI_ADDR
                                                           .EXTRN   MOM$SRVOPEN, MOM$SRVWRITE
                                                           .EXTRN   MOM$SRVREWIND, MOM$SRVCLOSE

                                                           .PSECT   $CODE$,NOWRT,2

                                      000C 00000           .ENTRY   MOM$OPERSERVICE, Save R2,R3                          ; 0149
                  6D       0085    CF   DE 00002           MOVAL    5$, (FP)                                             ; 0173
       00000000G  EF               00   FB 00007           CALLS    #0, MOM$PARSE_NICE_ENTITY                            ; 0189
                  50 00000000G     EF   9A 0000E           MOVZBL   MOM$GB_FUNCTION, R0                                  ; 0193
                  0F               50   91 00015           CMPB     R0, #15                                              ; 0195
                                   10   12 00018           BNEQ     1$                                                   ;
                  53 00000000G     EF   9E 0001A           MOVAB    MOM$LOAD, MOM_A_ROUTINE                              ; 0197
                  52 00000000G     EF   9E 00021           MOVAB    MOM$NPA_LOAD, MOM_A_NPATBL                           ; 0198
                                   2C   11 00028           BRB      4$                                                   ; 0193
                  11               50   91 0002A 1$:       CMPB     R0, #17                                              ; 0201
                                   10   12 0002D           BNEQ     2$                                                   ;
                  53 00000000V     EF   9E 0002F           MOVAB    MOM$TRIGGER, MOM_A_ROUTINE                           ; 0203
                  52 00000000G     EF   9E 00036           MOVAB    MOM$NPA_TRIGGER, MOM_A_NPATBL                        ; 0204
                                   17   11 0003D           BRB      4$                                                   ; 0193
                  12               50   91 0003F 2$:       CMPB     R0, #18                                              ; 0207
                                   08   12 00042           BNEQ     3$                                                   ;
       00000000G  EF               00   FB 00044           CALLS    #0, MOM$TEST                                         ; 0209
                                        04 0004B           RET                                                           ; 0208
                  7E               01   CE 0004C 3$:       MNEGL    #1, -(SP)                                            ; 0215
       00000000G  EF               01   FB 0004F           CALLS    #1, MOM$ERROR                                        ;
                                   52   DD 00056 4$:       PUSHL    MOM_A_NPATBL                                         ; 0226
                     00000000G     EF   9F 00058           PUSHAB   MOM$AB_NPARSE_BLK                                    ;
       00000000G  EF               02   FB 0005E           CALLS    #2, NMA$NPARSE                                       ;
       00000000G  EF               07   90 00065           MOVB     #7, MOM$GW_EVT_CODE                                  ; 0230
       00000000G  EF               04   90 0006C           MOVB     #4, MOM$GB_EVT_PRSN                                  ; 0231
       00000000G  EF               00   FB 00073           CALLS    #0, MOM$GETSRVDATA                                   ; 0233
       00000000G  EF               00   FB 0007A           CALLS    #0, MOM$GETSRVTIMER                                  ; 0238
                     00000000G     EF   9F 00081           PUSHAB   MOM$AB_CIB                                           ; 0244
                  63               01   FB 00087           CALLS    #1, (MOM_A_ROUTINE)                                  ;
                                        04 0008A           RET                                                           ; 0246
                                      0000 0008B 5$:       .WORD    Save nothing                                         ; 0173
                                   7E   D4 0008D           CLRL     -(SP)                                                ;
                                   5E  DD 0008F         PUSHL   SP                                         ;
                       7E      04  AC  7D 00091         MOVQ    4(AP), -(SP)                               ;
            00000000V  EF          03  FB 00095         CALLS   #3, MOM$SERVICEHANDLER                     ;
                                       04 0009C         RET                                                ;

; Routine Size:  157 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL 'mom$autoservice  Automatic maintenance operations request'
1 GLOBAL ROUTINE mom$autoservice : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine performs initialization and dispatching for all
1 !       automatic service requests.
1 !
1 ! IMPLICIT OUTPUTS:
1 !       MOM$GL_SERVICE_FLAGS [MOM$V_AUTOSERVICE] has a value of TRUE.
1 !       The Service Data (SVD) table contains the service parameter
1 !               information from the volatile data base and from the MOP message.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !       Signal errors.
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2     ptr,
2     rcv_mop_msg_dsc: VECTOR [2],
2     status;
2
2 !
2 ! Enable condition handler for cleanup.
2 !
2 enable mom$autohandler;
2 mom$ab_cib [cib$l_chan] = 0;
2 !
2 ! Set up the event in case of an aborted service request.
2 !
2 mom$gw_evt_code = evc$c_nma_abs;          ! Event code (aborted service request)
2 mom$gb_evt_prsn = evc$c_nma_prsn_loe;     ! Circuit open error
2 !
2 ! Get the service data base.
2 !
2 mom$gb_entity_code = mom$c_circuit;
2 !
2 ! Get the service line timer (used to time all interactions with the
2 ! target node).
2 !
2 mom$getsrvtimer ();
2 !
2 ! Open a MOP channel on the circuit specified in SYS$NET (translated
2 ! into MOM$GQ_ENTITY_BUF_DSC).
2 !
2 mom$mopopen (mom$ab_cib [cib$l_chan]);
2 !
2 ! MOM is started up for autoservice when this node receives a MOP
2 ! message from some other node which requests a maintenance operation.
2 ! Get this MOP message and parse it.
2 !
2 mom_get_initial_mop_msg (rcv_mop_msg_dsc);
2 !
```

```
;  307          0304 2 ! Get the service data and timer from the volatile data base entry for
;  308          0305 2 ! this device.  For NI circuits, this uses the NI address obtained by
;  309          0306 2 ! MOM_GET_INITIAL_MOP_MSG.
;  310          0307 2 !
;  311          0308 2 mom$getsrvdata ();
;  312          0309 2 !
;  313          0310 2 ! Set up Channel Information Block for channel.  For NI circuits, this
;  314          0311 2 ! sets up the NI protocol for the circuit, and associate it with a specific
;  315          0312 2 ! NI destination.
;  316          0313 2 !
;  317          0314 2 mom$init_CIB (mom$ab_cib,                    ! Channel Information Block addr
;  318          0315 2                 .mom$gb_function,           ! Function = load, dump, test
;  319          0316 2                 svd$gk_pcno_pha,            ! NI Phycial Address
;  320          0317 2                 svd$gk_pcno_add,            ! Node address
;  321          0318 2                 svd$gk_pcno_hwa);           ! NI hardware address
;  322          0319 2 !
;  323          0320 2 ! For load and dump:
;  324          0321 2 ! If the MOP message sent by the target an NI circuit multicast and
;  325          0322 2 ! it's not a request for the secondary loader then volunteer assistance.
;  326          0323 2 ! The target will perform the load or dump sequence with the first host
;  327          0324 2 ! to respond to the multicast.
;  328          0325 2 !
;  329          0326 2 IF .mom$gl_service_flags [mom$v_ni_volunteering] AND
;  330          0327 3    NOT ((.mom$gb_function EQL nma$c_fnc_loa) AND
;  331          0328 4         (.mom$ab_service_data [svd$gk_pcno_sty, svd$l_param] EQL
;  332          0329 2                                              nma$c_soft_secl)) THEN
;  333          0330 2    mom_volunteer_assistance (rcv_mop_msg_dsc);
;  334          0331 2
;  335          0332 2 !
;  336          0333 2 ! Dispatch the autoservice function to the correct routine to handle it.
;  337          0334 2 ! (The service routine will signal status and the condition handler will
;  338          0335 2 ! take care of cleanup.)
;  339          0336 2 !
;  340          0337 2
;  341          0338 2 SELECTONEU .mom$gb_function OF
;  342          0339 2     SET
;  343          0340 2     [nma$c_fnc_tes]: mom$passiveloop (rcv_mop_msg_dsc);
;  344          0341 2
;  345          0342 2     [nma$c_fnc_loa]: mom$load (mom$ab_cib);
;  346          0343 2
;  347          0344 2     [nma$c_fnc_dum]: mom$dump ();
;  348          0345 2
;  349          0346 2     [OTHERWISE]:     mom$error (nma$c_sts_fun);
;  350          0347 2
;  351          0348 2     TES;
;  352          0349 1 END;                                ! End of MOM$AUTOSERVICE





                                  001C 00000          .ENTRY  MOM$AUTOSERVICE, Save R2,R3,R4              ; 0248
                  54 00000000G  EF   9E 00002          MOVAB   MOM$GB_FUNCTION, R4                         ;
                  53 00000000G  EF   9E 00009          MOVAB   MOM$AB_CIB, R3                              ;
                  5E              08   C2 00010          SUBL2   #8, SP                                      ;
                  6D      00AD    CF   DE 00013          MOVAL   6$, (FP)                                    ; 0266
                                  63   D4 00018          CLRL    MOM$AB_CIB                                  ; 0277
```

```
          00000000G  EF            07  90 0001A          MOVB     #7, MOM$GW_EVT_CODE                                ; 0281
          00000000G  EF            04  90 00021          MOVB     #4, MOM$GB_EVT_PRSN                                ; 0282
          00000000G  EF            02  90 00028          MOVB     #2, MOM$GB_ENTITY_CODE                             ; 0286
          00000000G  EF            00  FB 0002F          CALLS    #0, MOM$GETSRVTIMER                                ; 0291
                                   53  DD 00036          PUSHL    R3                                                 ; 0296
          00000000G  EF            01  FB 00038          CALLS    #1, MOM$MOPOPEN                                    ;
                                   5E  DD 0003F          PUSHL    SP                                                 ; 0302
          00000000V  EF            01  FB 00041          CALLS    #1, MOM_GET_INITIAL_MOP_MSG                        ;
          00000000G  EF            00  FB 00048          CALLS    #0, MOM$GETSRVDATA                                 ; 0308
                       00000000G   8F  DD 0004F          PUSHL    #SVD$GK_PCNO_HWA                                   ; 0314
                       00000000G   8F  DD 00055          PUSHL    #SVD$GK_PCNO_ADD                                   ;
                       00000000G   8F  DD 0005B          PUSHL    #SVD$GK_PCNO_PHA                                   ;
                     7E            64  9A 00061          MOVZBL   MOM$GB_FUNCTION, -(SP)                             ; 0315
                                   53  DD 00064          PUSHL    R3                                                 ; 0314
          00000000G  EF            05  FB 00066          CALLS    #5, MOM$INIT_CIB                                   ;
                       00000000G   EF  95 0006D          TSTB     MOM$GL_SERVICE_FLAGS                               ; 0326
                                   16  18 00073          BGEQ     2$                                                 ;
                     0F            64  91 00075          CMPB     MOM$GB_FUNCTION, #15                               ; 0327
                                   08  12 00078          BNEQ     1$                                                 ;
                       00000000*   EF  D5 0007A          TSTL     <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_STY*137>-; 0328
                                                                  >+9>                                               ;
                                   09  13 00080          BEQL     2$                                                 ;
                                   5E  DD 00082 1$:      PUSHL    SP                                                 ; 0330
          00000000V  EF            01  FB 00084          CALLS    #1, MOM_VOLUNTEER_ASSISTANCE                       ;
                     52            64  9A 0008B 2$:      MOVZBL   MOM$GB_FUNCTION, R2                                ; 0338
                     12            52  91 0008E          CMPB     R2, #18                                            ; 0340
                                   0A  12 00091          BNEQ     3$                                                 ;
                                   5E  DD 00093          PUSHL    SP                                                 ;
          00000000G  EF            01  FB 00095          CALLS    #1, MOM$PASSIVELOOP                                ;
                                       04 0009C          RET                                                         ;
                     0F            52  91 0009D 3$:      CMPB     R2, #15                                            ; 0342
                                   0A  12 000A0          BNEQ     4$                                                 ;
                                   53  DD 000A2          PUSHL    R3                                                 ;
          00000000G  EF            01  FB 000A4          CALLS    #1, MOM$LOAD                                       ;
                                       04 000AB          RET                                                         ;
                     10            52  91 000AC 4$:      CMPB     R2, #16                                            ; 0344
                                   08  12 000AF          BNEQ     5$                                                 ;
          00000000V  EF            00  FB 000B1          CALLS    #0, MOM$DUMP                                       ;
                                       04 000B8          RET                                                         ;
                     7E            01  CE 000B9 5$:      MNEGL    #1, -(SP)                                          ; 0346
          00000000G  EF            01  FB 000BC          CALLS    #1, MOM$ERROR                                      ;
                                       04 000C3          RET                                                         ; 0349
                                     0000 000C4 6$:      .WORD    Save nothing                                       ; 0266
                                   7E  D4 000C6          CLRL     -(SP)                                              ;
                                   5E  DD 000C8          PUSHL    SP                                                 ;
                     7E        04  AC  7D 000CA          MOVQ     4(AP), -(SP)                                       ;
          00000000V  EF            03  FB 000CE          CALLS    #3, MOM$AUTOHANDLER                                ;
                                       04 000D5          RET                                                         ;

; Routine Size:  214 bytes,    Routine Base:  $CODE$ + 009D
```

```
1 %SBTTL 'mom_get_initial_mop_msg     Get autoservice initial MOP message'
1 ROUTINE mom_get_initial_mop_msg (rcv_mop_msg_dsc) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !       MOM is started up for autoservice when this node receives a MOP
1 !       message from some other node which requests a maintenance operation.
1 !       This routine is called by MOM$AUTOSERVICE to get the MOP message
1 !       from the device driver and to parse it.
1 !
1 ! INPUTS:
1 !       RCV_MOP_MSG_DSC - Address at which to put a descriptor of the
1 !               initial MOP message.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     rcv_mop_msg_dsc: REF VECTOR;
2
2 LOCAL
2     status,
2     iosb: $iosb,
2     msgsize,
2     send_mop_msg_dsc   : VECTOR [2],
2     ni_header : BBLOCK [nih$k_ni_header_len];
2
2 !
2 ! Initialize the MOP message descriptor.
2 !
2 rcv_mop_msg_dsc [0] = .mom$gq_mop_rcv_buf_dsc [0];
2 rcv_mop_msg_dsc [1] = .mom$gq_mop_rcv_buf_dsc [1];
2 !
2 ! If the driver is holding a MOP message, issue a read to get it.
2 !
2 status = mom$mop_receive_qiow (.mom$ab_cib [cib$l_chan],
2                                  .rcv_mop_msg_dsc,
2                                  ni_header);
2 IF .status THEN
3     BEGIN
3     !
3     ! Check the NI header to see if the target sent the MOP message to the
3     ! NI multicast address (meaning that any NI node enabled for multicast
3     ! can respond to the message.)  If so, MOM must respond by volunteering
3     ! to perform the operation requested.
3     !
3     IF .mom$gl_service_flags [mom$v_ni_circ] THEN
4         BEGIN
4         !
4         ! Dump out the NI header to debug logging.
4         !
```

```
;  411       0407 4          mom$debug_msg (dbg$c_mopio,
;  412       0408 4                 ni_header,
;  413       0409 4                 nih$k_ni_header_len,
;  414       0410 4                 $ASCID ('Header on NI message initiating autoservice'));
;  415       0411 4          IF .ni_header [nih$b_multicast] THEN
;  416       0412 5              BEGIN
;  417       0413 5              mom$gl_service_flags [mom$v_ni_volunteering] = true;
;  418       0414 5              mom$gl_service_flags [mom$v_ni_multicast] = true;
;  419       0415 4              END;
;  420       0416 4          !
;  421       0417 4          ! Save the source and destination addresses of the MOP message
;  422       0418 4          ! which initiated autoservice.
;  423       0419 4          !
;  424       0420 4          CH$MOVE (6, ni_header [nih$t_source_ni_addr],
;  425       0421 4                  mom$ab_service_data [svd$gk_pcno_pha, svd$t_string]);
;  426       0422 4          mom$ab_service_data [svd$gk_pcno_pha, svd$b_string_len] = 6;
;  427       0423 4          mom$ab_service_data [svd$gk_pcno_pha, svd$v_msg_param] = true;
;  428       0424 4          CH$MOVE (6, ni_header [nih$t_dest_ni_addr],
;  429       0425 4                  mom$ab_service_data [svd$gk_pcno_$da, svd$t_string]);
;  430       0426 4          mom$ab_service_data [svd$gk_pcno_$da, svd$b_string_len] = 6;
;  431       0427 4          mom$ab_service_data [svd$gk_pcno_$da, svd$v_msg_param] = true;
;  432       0428 3          END;
;  433       0429 3      !
;  434       0430 3      ! Parse the MOP message that was received.
;  435       0431 3      !
;  436       0432 3      mom$ab_nparse_blk [npa$l_msgcnt] = .rcv_mop_msg_dsc [0];
;  437       0433 3      mom$ab_nparse_blk [npa$l_msgptr] = .rcv_mop_msg_dsc [1];
;  438       0434 3      status = nma$nparse (mom$ab_nparse_blk, mom$npa_mopinit);
;  439       0435 3      END
;  440       0436 2  ELSE
;  441       0437 2  !
;  442       0438 2  ! If no MOP message was returned by the driver, send a bad MOP message to
;  443       0439 2  ! get the target to retransmit the MOP message which caused me to be
;  444       0440 2  ! started up.  This code will go away when all the drivers have been
;  445       0441 2  ! changed to save that MOP message and return it to MOM$MOPOPEN.
;  446       0442 2  ! Note that this code path is taken for point-to-point and multipoint
;  447       0443 2  ! lines.
;  448       0444 2  !
;  449       0445 3      BEGIN
;  450       0446 3      !
;  451       0447 3      ! Initialize the MOP message descriptor.
;  452       0448 3      !
;  453       0449 3      rcv_mop_msg_dsc [0] = .mom$gq_mop_rcv_buf_dsc [0];
;  454       0450 3      rcv_mop_msg_dsc [1] = .mom$gq_mop_rcv_buf_dsc [1];
;  455       0451 3      !
;  456       0452 3      ! Build the bad message (old MOP mode running, new Request dump
;  457       0453 3      ! service) to send.
;  458       0454 3      !
;  459       0455 3      mom$bldmoprds (send_mop_msg_dsc);
;  460       0456 3      mom$gb_evt_prsn = evc$c_nma_prsn_err; ! Receive error reason
;  461       0457 3      !
;  462       0458 3      ! Send the bad MOP mode running message (used to get the target to
;  463       0459 3      ! retransmit the last thing it sent - this is done for device drivers
;  464       0460 3      ! that lose the MOP message that causes MOM to get started up.)
;  465       0461 3      !
;  466       0462 3      mom$ab_cib [cib$l_retry_cnt] = 5;
;  467       0463 3      DECR i FROM 5 TO 0 DO
;  468             0464  4           BEGIN
;  469             0465  4           status = mom$mopsndrcv (mom$ab_cib, send_mop_msg_dsc,
;  470             0466  4                                   mom$ab_cib, .rcv_mop_msg_dsc,
;  471             0467  4                                   rcv_mop_msg_dsc [0],
;  472             0468  4                                   0);     ! Don't skip program load requests
;  473             0469  4           mom$chk_mop_error (.status);
;  474             0470  4           !
;  475             0471  4           ! Parse the MOP message that was received.
;  476             0472  4           !
;  477             0473  4           mom$ab_nparse_blk [npa$l_msgcnt] = .rcv_mop_msg_dsc [0];
;  478             0474  4           mom$ab_nparse_blk [npa$l_msgptr] = .rcv_mop_msg_dsc [1];
;  479             0475  4           status = nma$nparse (mom$ab_nparse_blk, mom$npa_mopinit);
;  480             0476  4           IF .status THEN
;  481             0477  4               EXITLOOP;
;  482             0478  3           END;
;  483             0479  2       END;
;  484             0480  2   !
;  485             0481  2   ! If a bad MOP message came from the target, give up.
;  486             0482  2   !
;  487             0483  2   IF NOT .status THEN
;  488             0484  3       BEGIN
;  489             0485  3       mom$bld_reply (mom$ab_msgblock, msgsize);
;  490             0486  3       $signal_msg (mom$ab_nice_xmit_buf, .msgsize);
;  491             0487  2       END;
;  492             0488  2
;  493             0489  1  END;                       !  of mom_get_initial_mop_msg


                                                                      .PSECT  $PLIT$,NOWRT,NOEXE,2

65 6D 20 49 4E 20 6E 6F 20 72 65 64 61 65 48  00000 P.AAB:  .ASCII  \Header on NI message initiating autoserv\  ;
6E 69 74 61 69 74 69 6E 69 20 65 67 61 73 73  0000F                                                            ;
            76 72 65 73 6F 74 75 61 20 67  0001E                                                            ;
                                     65 63 69  00028         .ASCII  \ice\                                   ;
                                               0002B         .BLKB   1                                       ;
                                    0000002B  0002C P.AAA:  .LONG   43                                      ;
                                    00000000' 00030         .ADDRESS P.AAB                                  ;


                                                                      .PSECT  $CODE$,NOWRT,2

                                          0FFC 00000 MOM_GET_INITIAL_MOP_MSG:
                                                             .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 0351
                    5B 00000000G   EF   9E 00002             MOVAB   MOM$NPA_MOPINIT, R11                    ;
                    5A 00000000G   EF   9E 00009             MOVAB   MOM$GL_SERVICE_FLAGS, R10               ;
                    59 00000000G   EF   9E 00010             MOVAB   MOM$AB_NPARSE_BLK+4, R9                 ;
                    58 00000000G   EF   9E 00017             MOVAB   MOM$AB_CIB, R8                          ;
                    5E             24   C2 0001E             SUBL2   #36, SP                                 ;
                    56          04 AC   D0 00021             MOVL    RCV_MOP_MSG_DSC, R6                     ; 0386
                    66 00000000G   EF   7D 00025             MOVQ    MOM$GQ_MOP_RCV_BUF_DSC, (R6)            ;
                                04 AE   9F 0002C             PUSHAB  NI_HEADER                               ; 0391
                                   56   DD 0002F             PUSHL   R6                                      ; 0392
                                   68   DD 00031             PUSHL   MOM$AB_CIB                              ; 0391
          00000000G   EF           03   FB 00033             CALLS   #3, MOM$MOP_RECEIVE_QIOW                ;
                      57           50   D0 0003A             MOVL    R0, STATUS                              ;
```

```
                        62              57  E9 0003D            BLBC    STATUS, 3$                                              ; 0394
              4A        6A              01  E1 00040            BBC     #1, MOM$GL_SERVICE_FLAGS, 2$                            ; 0402
                              00000000' EF  9F 00044            PUSHAB  P.AAA                                                   ; 0410
                                        0E  DD 0004A            PUSHL   #14                                                     ; 0407
                                  0C    AE  9F 0004C            PUSHAB  NI_HEADER                                               ;
                                        05  DD 0004F            PUSHL   #5                                                      ;
                   00000000G    EF      04  FB 00051            CALLS   #4, MOM$DEBUG_MSG                                       ;
                        04        04    AE  E9 00058            BLBC    NI_HEADER, 1$                                           ; 0411
                        6A        A0    8F  88 0005C            BISB2   #160, MOM$GL_SERVICE_FLAGS                              ; 0414
    00000000*   EF        0A    AE      06  28 00060 1$:        MOVC3   #6, NI_HEADER+6, <<MOM$AB_SERVICE_DATA+-                ; 0421
                                                                        <SVD$GK_PCNO_PHA*137>>+9>                               ;
                   00000000*    EF      06  90 00069            MOVB    #6, <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_PHA*-            ; 0422
                                                                        137>>+8>                                                ;
                   00000000*    EF      01  88 00070            BISB2   #1, <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_PHA*-            ; 0423
                                                                        137>>+7>                                                ;
    00000000*   EF        04    AE      06  28 00077            MOVC3   #6, NI_HEADER, <<MOM$AB_SERVICE_DATA+-                  ; 0425
                                                                        <SVD$GK_PCNO_$DA*137>>+9>                               ;
                   00000000*    EF      06  90 00080            MOVB    #6, <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_$DA*-            ; 0426
                                                                        137>>+8>                                                ;
                   00000000*    EF      01  88 00087            BISB2   #1, <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_$DA*-            ; 0427
                                                                        137>>+7>                                                ;
                        69              66  7D 0008E 2$:        MOVQ    (R6), MOM$AB_NPARSE_BLK+4                               ; 0432
                                        5B  DD 00091            PUSHL   R11                                                     ; 0434
                                  FC    A9  9F 00093            PUSHAB  MOM$AB_NPARSE_BLK                                       ;
                   00000000G    EF      02  FB 00096            CALLS   #2, NMA$NPARSE                                          ;
                                57      50  D0 0009D            MOVL    R0, STATUS                                              ;
                                57      11  000A0               BRB     5$                                                      ; 0394
                        66 00000000G    EF  7D 000A2 3$:        MOVQ    MOM$GQ_MOP_RCV_BUF_DSC, (R6)                            ; 0449
                                  14    AE  9F 000A9            PUSHAB  SEND_MOP_MSG_DSC                                        ; 0455
                   00000000G    EF      01  FB 000AC            CALLS   #1, MOM$BLDMOPRDS                                       ;
                   00000000G    EF      01  90 000B3            MOVB    #1, MOM$GB_EVT_PRSN                                     ; 0456
                          12    A8      05  D0 000BA            MOVL    #5, MOM$AB_CIB+18                                       ; 0462
                                52      05  D0 000BE            MOVL    #5, I                                                   ; 0463
                                        7E  D4 000C1 4$:        CLRL    -(SP)                                                   ; 0467
                                        56  DD 000C3            PUSHL   R6                                                      ;
                                        56  DD 000C5            PUSHL   R6                                                      ;
                                        58  DD 000C7            PUSHL   R8                                                      ; 0465
                                  24    AE  9F 000C9            PUSHAB  SEND_MOP_MSG_DSC                                        ;
                                        58  DD 000CC            PUSHL   R8                                                      ;
                   00000000G    EF      06  FB 000CE            CALLS   #6, MOM$MOPSNDRCV                                       ; 0467
                                57      50  D0 000D5            MOVL    R0, STATUS                                              ;
                                        57  DD 000D8            PUSHL   STATUS                                                  ; 0469
                   00000000V    EF      01  FB 000DA            CALLS   #1, MOM$CHK_MOP_ERROR                                   ;
                                69      66  7D 000E1            MOVQ    (R6), MOM$AB_NPARSE_BLK+4                               ; 0473
                                        5B  DD 000E4            PUSHL   R11                                                     ; 0475
                                  FC    A9  9F 000E6            PUSHAB  MOM$AB_NPARSE_BLK                                       ;
                   00000000G    EF      02  FB 000E9            CALLS   #2, NMA$NPARSE                                          ;
                                57      50  D0 000F0            MOVL    R0, STATUS                                              ;
                                2A      57  E8 000F3            BLBS    STATUS, 6$                                              ; 0476
                                C8      52  F4 000F6            SOBGEQ  I, 4$                                                   ; 0463
                                24      57  E8 000F9 5$:        BLBS    STATUS, 6$                                              ; 0483
                                        5E  DD 000FC            PUSHL   SP                                                      ; 0485
                              00000000G EF  9F 000FE            PUSHAB  MOM$AB_MSGBLOCK                                         ;
                   00000000G    EF      02  FB 00104            CALLS   #2, MOM$BLD_REPLY                                       ;
                                        6E  DD 0010B            PUSHL   MSGSIZE                                                 ; 0486
                              00000000G EF  9F 0010D            PUSHAB  MOM$AB_NICE_XMIT_BUF                                    ;
                              02070000  8F  DD 00113            PUSHL   #34013184                                               ;
                        0000000G  00            03  FB 00119        CALLS   #3, LIB$SIGNAL                                  ;
                                                    04 00120 6$:    RET                                                     ; 0489

; Routine Size:  289 bytes,     Routine Base:  $CODE$ + 0173
```

```
1 %SBTTL 'mom_volunteer_assistance   Volunteer Assistance'
1 ROUTINE mom_volunteer_assistance (rcv_mop_msg_dsc) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !       If there is a node on the NI whose host is down and it needs a
1 !       down line load or an up line dump, it will broadcast a request
1 !       for to the NI multicast address for the dump or load.  If the
1 !       request is not for a secondary loader (in which case MOM responds
1 !       with a secondary loader), MOM sends a volunteer assistance to the
1 !       target node.  The target will chose the first host to volunteer to
1 !       continue the load or dump sequence with.  If a response is
1 !       received to the volunteer assistance sent here, then this node
1 !       was chosen by the target.  Parse the message and return to
1 !       perform the operation.
1 !
1 ! INPUTS:
1 !       None
1 !
1 ! IMPLICIT OUTPUTS:
1 !       Writes an event indicating that assistance was volunteered.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !--
1
2 BEGIN
2
2 MAP
2     rcv_mop_msg_dsc :   REF VECTOR;
2
2 LOCAL
2     status,
2     msgsize;
2
2 !
2 ! Initialize the MOP message descriptor.
2 !
2 rcv_mop_msg_dsc [0] = .mom$gq_mop_rcv_buf_dsc [0];
2 rcv_mop_msg_dsc [1] = .mom$gq_mop_rcv_buf_dsc [1];
2 !
2 ! Sent assistance volunteer to node requesting assistance.  Send it once if
2 ! the correct destination NI address is available.  Otherwise, try once to the
2 ! target's HIORD address and once to its hardware address.
2 !
2 IF (NOT .mom$ab_cib [cib$v_target_addr_fixed]) THEN
2     mom$ab_cib [cib$l_retry_cnt] = 2
2 ELSE
2     mom$ab_cib [cib$l_retry_cnt] = 1;
2 status = mom$mopsndrcv (mom$ab_cib, UPLIT LONG (1,
2                                                    UPLIT BYTE (mop$_fct_asv)),
2                         mom$ab_cib, .rcv_mop_msg_dsc,
2                         rcv_mop_msg_dsc [0],
2                         mom$k_skip_multicasts); ! Ignore received multicasts.
2
2 mom$chk_mop_error (.status);
```

```
;   552         0547  2 !
;   553         0548  2 ! In case a response to the volunteer was received, restore the retry count
;   554         0549  2 ! for normal operations.
;   555         0550  2 !
;   556         0551  2 mom$ab_cib [cib$l_retry_cnt] = 5;
;   557         0552  2 !
;   558         0553  2 ! Parse the MOP message that was received.
;   559         0554  2 !
;   560         0555  2 mom$ab_nparse_blk [npa$l_msgcnt] = .rcv_mop_msg_dsc [0];
;   561         0556  2 mom$ab_nparse_blk [npa$l_msgptr] = .rcv_mop_msg_dsc [1];
;   562         0557  2 status = nma$nparse (mom$ab_nparse_blk, mom$npa_mopinit);
;   563         0558  2 !
;   564         0559  2 ! If a bad MOP message came from the target, give up.
;   565         0560  2 !
;   566         0561  2 IF NOT .status THEN
;   567         0562  3     BEGIN
;   568         0563  3     mom$bld_reply (mom$ab_msgblock, msgsize);
;   569         0564  3     $signal_msg (mom$ab_nice_xmit_buf, .msgsize);
;   570         0565  2     END;
;   571         0566  2 mom$gl_service_flags [mom$v_ni_volunteering] = false;
;   572         0567  2 !
;   573         0568  2 ! Don't log any events for a volunteer assistance that gets no response.
;   574         0569  2 ! This is because every node on the NI that's enabled for multicast would
;   575         0570  2 ! log this event for every multicast request.  That's generating mucho
;   576         0571  2 ! not very interesting data.
;   577         0572  2 !
;   578         0573  2 RETURN .status;
;   579         0574  1 END;                          ! of mom_volunteer_assistance


                                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

                                          03  00034 P.AAD:    .BYTE    3                                          ;
                                              00035           .BLKB    3
                                    00000001  00038 P.AAC:    .LONG    1                                          ;
                                   00000000'  0003C           .ADDRESS P.AAD                                      ;


                                                              .PSECT  $CODE$,NOWRT,2

                                        001C  00000 MOM_VOLUNTEER_ASSISTANCE:
                                                              .WORD    Save R2,R3,R4                              ; 0491
                      54 00000000G   EF   9E  00002           MOVAB    MOM$AB_CIB+18, R4                          ;
                      5E             04   C2  00009           SUBL2    #4, SP                                     ;
                      52        04   AC   D0  0000C           MOVL     RCV_MOP_MSG_DSC, R2                        ; 0529
                      62 00000000G   EF   7D  00010           MOVQ     MOM$GQ_MOP_RCV_BUF_DSC, (R2)               ;
                      05        FE   A4   E8  00017           BLBS     MOM$AB_CIB+16, 1$                          ; 0536
                      64             02   D0  0001B           MOVL     #2, MOM$AB_CIB+18                          ; 0537
                                     03   11  0001E           BRB      2$                                         ;
                      64             01   D0  00020 1$:       MOVL     #1, MOM$AB_CIB+18                          ; 0539
                      7E             01   CE  00023 2$:       MNEGL    #1, -(SP)                                  ; 0543
                                     52   DD  00026           PUSHL    R2                                         ;
                                     52   DD  00028           PUSHL    R2                                         ;
                                EE   A4   9F  0002A           PUSHAB   MOM$AB_CIB                                 ; 0540
                         00000000'   EF   9F  0002D           PUSHAB   P.AAC                                      ;
```

```
                                   EE   A4  9F 00033          PUSHAB   MOM$AB_CIB                                    ;
                  00000000G  EF         06  FB 00036          CALLS    #6, MOM$MOPSNDRCV                             ; 0543
                             53         50  D0 0003D          MOVL     R0, STATUS                                    ;
                                        53  DD 00040          PUSHL    STATUS                                        ; 0546
                  00000000V  EF         01  FB 00042          CALLS    #1, MOM$CHK_MOP_ERROR                         ;
                             64         05  D0 00049          MOVL     #5, MOM$AB_CIB+18                             ; 0551
                  00000000G  EF         62  7D 0004C          MOVQ     (R2), MOM$AB_NPARSE_BLK+4                     ; 0555
                                 00000000G  EF  9F 00053      PUSHAB   MOM$NPA_MOPINIT                               ; 0557
                                 00000000G  EF  9F 00059      PUSHAB   MOM$AB_NPARSE_BLK                             ;
                  00000000G  EF         02  FB 0005F          CALLS    #2, NMA$NPARSE                                ;
                             53         50  D0 00066          MOVL     R0, STATUS                                    ;
                             24         53  E8 00069          BLBS     STATUS, 3$                                    ; 0561
                                        5E  DD 0006C          PUSHL    SP                                            ; 0563
                                 00000000G  EF  9F 0006E      PUSHAB   MOM$AB_MSGBLOCK                               ;
                  00000000G  EF         02  FB 00074          CALLS    #2, MOM$BLD_REPLY                             ;
                                        6E  DD 0007B          PUSHL    MSGSIZE                                       ; 0564
                                 00000000G  EF  9F 0007D      PUSHAB   MOM$AB_NICE_XMIT_BUF                          ;
                                 02070000   8F  DD 00083      PUSHL    #34013184                                     ;
                  00000000G  00         03  FB 00089          CALLS    #3, LIB$SIGNAL                                ;
                  00000000G  EF    80   8F  8A 00090  3$:     BICB2    #128, MOM$GL_SERVICE_FLAGS                    ; 0566
                                            04 00098          RET                                                    ; 0574

; Routine Size:  153 bytes,     Routine Base:  $CODE$ + 0294
```

```
1 %SBTTL 'mom$dump  Upline dump'
1 GLOBAL ROUTINE mom$dump : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine performs the upline system dump operation.
1 !
1 ! IMPLICIT INPUTS:
1 !       MOM$AB_CIB = QIO channel information block for MOP messages.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !       Signal errors.
1 !
1 !--
1
2 BEGIN
2
2 MACRO
2                                                   ! MOP Request Memory Dump msg
2     rmd_b_fct           = 0,0,8,0%,               !  Function code
2     rmd_l_memaddr       = 1,0,32,0%,              !  Memory address
2     rmd_w_numlocs       = 5,0,16,0%,              !  Number of locations
2                                                   ! MOP Memory Dump Data message
2     mdd_b_fct   = 0,0,8,0%,                       !  Function code
2     mdd_l_memaddr       = 1,0,32,0%,              !  Memory address
2     mdd_t_data  = 5,0,0,0%;                       !  Data
2
2 LITERAL
2     mdd_c_datsiz        = 256,                    ! MOP Memory Dump data size
2     mdd_c_hdrsiz        = 5,                      ! MOP Memory Dump header size
2     recsiz              = 512;                    ! Record buffer size
2
2 LOCAL
2     len,adr,                                      ! Temp length and address
2     retries,                                      ! Retries remaining count
2     memsiz,                                       ! Memory size
2     memadr,                                       ! Memory address
2     curmemadr,                                    ! Current memory address
2     fildsc : VECTOR [2],                          ! File name descriptor
2     rmdbuf : BBLOCK [7],                          ! Request Memory Dump message
2     rmddsc : VECTOR [2],                          ! Request Memory Dump desc
2     mddlen,                                       ! Memory Dump data length
2     mddbuf : BBLOCK [mdd_c_hdrsiz                 ! Memory Dump Data message
2                       + mdd_c_datsiz],
2     mdddsc : VECTOR [2],                          ! Memory Dump Data descriptor
2     recidx,                                       ! Record buffer index
2     recbuf : VECTOR [recsiz,BYTE],                ! Record buffer
2     recdsc : VECTOR [2],                          ! Record buffer descriptor
2
2     snddsc : VECTOR [2],
2     msgdsc : VECTOR [2],
2     skip_msg_dsc_addr,                            ! Address of descriptor of
2                                                   ! MOP message to skip over.
2     msgsize,
2           status;
2       !
2       ! Set the circuit substate and log the automatic service event for
2       ! upline dump.
2       !
2       mom$gw_evt_code = evc$c_nma_als;
2       mom$gb_evt_pser = evc$c_nma_pser_dum;
2       mom$log_event (0, 0);
2
2       mom$mopsetsubstate (nma$c_linss_adu,
2                           .mom$ab_cib [cib$l_chan]);! -AUTODUMPING
2       !
2       ! Get the output file name from the data base.
2       !
2       fildsc [1] = 0;
2       IF .mom$ab_service_data [svd$gk_pcno_dum, svd$b_string_len] LEQ 0 THEN
3           BEGIN
3           mom$error (nma$c_sts_pms, nma$c_pcno_dum);
3           RETURN;
3           END
2       ELSE
3           BEGIN
3           fildsc [0] = .mom$ab_service_data [svd$gk_pcno_dum, svd$b_string_len];
3           fildsc [1] = mom$ab_service_data [svd$gk_pcno_dum, svd$t_string];
2           END;
2       !
2       ! Get the memory size of the system being dumped
2       !
2       memsiz = .mom$ab_service_data [svd$gk_pcno_dct, svd$l_param];
2       IF .memsiz LEQ -1 THEN
3           BEGIN
3           mom$error (nma$c_sts_pms, nma$c_pcno_dct);
3           RETURN;
2           END;
2
2       !
2       ! Get the starting memory address of the system being dumped
2       !
2       memadr = .mom$ab_service_data [svd$gk_pcno_dad, svd$l_param];
2       IF .memadr LEQ -1 THEN
2           memadr = 0;
2
2       !
2       ! Open the output file.
2       !
2       status = mom$srvopen (fildsc, nma$c_opn_ac_rw);
2
2       IF NOT .status THEN
3           BEGIN
3           mom$ab_msgblock [msb$w_detail] = nma$c_fopdtl_dfl;
3           mom$bld_reply (mom$ab_msgblock, msgsize);
3           $signal_msg (mom$ab_nice_xmit_buf, .msgsize);
3           RETURN;
3           END;
2       !
2       ! Output the trace message.
2       !
```

```
2 mom$debug_txt (dbg$c_srvtrc, $ASCID ('Dumping'));
2
2 rmdbuf [rmd_b_fct] = mop$_fct_rmd;
2 rmdbuf [rmd_w_numlocs] = mdd_c_datsiz;
2 rmddsc [0] = %ALLOCATION(rmdbuf);
2 rmddsc [1] = rmdbuf;
2
2 mdddsc [0] = %ALLOCATION(mddbuf);
2 mdddsc [1] = mddbuf;
2 !
2 ! Dump the remote system's memory and retry if it fails.
2 !
2 skip_msg_dsc_addr = mom$gq_mop_msg_dsc;
2 curmemadr = .memadr;
2 recidx = 0;
2 retries = 2;
2
2 WHILE .curmemadr LSSU .memsiz
2       AND .retries GTRU 0
2       DO
3       BEGIN
3       !
3       ! Send 'Request Memory Dump' message and receive a response message.
3       ! Skip over retransmitted Request Memory Dump messages from the target
3       ! the first time through.
3       !
3       rmdbuf [rmd_l_memaddr] = .curmemadr;
3       STATUS = mom$mopsndrcv (mom$ab_cib, rmddsc,
3                               mom$ab_cib, mdddsc,
3                               mddlen,
3                               .skip_msg_dsc_addr); ! Skip duplicate dump requests.
3       skip_msg_dsc_addr = 0;
3       !
3       ! If the response message is a 'Memory Dump Data' and the memory
3       ! address is the one requested, copy the data to the record buffer.
3       !
3       IF .status
4          AND (.mddlen GEQU mdd_c_hdrsiz)
4          AND (.mddbuf [mdd_b_fct] EQL mop$_fct_mdd)
3          AND (.mddbuf [mdd_l_memaddr] EQL .curmemadr) THEN
4           BEGIN
4           CH$MOVE (.mddlen - mdd_c_hdrsiz, mddbuf [mdd_t_data],
4                   recbuf [.recidx]);
4           recidx = (.recidx + mdd_c_datsiz) MOD recsiz;
4
4           !
4           ! If the record buffer was just filled, write it to the file.
4           !
4           IF .recidx EQL 0 THEN
4              mom$srvwrite (recbuf, recsiz);
4
4           curmemadr = .curmemadr + mdd_c_datsiz;
4           END
3       ELSE
4           BEGIN
4           !
4           ! The response message isn't correct, so tell the dumper to
```

```
4         ! start dump again by sending a bad (opcode only) 'Request Dump
4         ! Service' MOP message.
4         !
4         mom$bldmoprds (snddsc);
4         msgdsc [1] = .mom$gq_mop_rcv_buf_dsc [1];
4
4         status = mom$mopsndrcv (mom$ab_cib, snddsc,
4                                 mom$ab_cib, mom$gq_mop_rcv_buf_dsc,
4                                 msgdsc [0],
4                                 0);              ! Don't skip program load requests
4         mom$chk_mop_error (.status);
4         mom$ab_nparse_blk [npa$l_msgcnt] = .msgdsc [0];
4         mom$ab_nparse_blk [npa$l_msgptr] = .msgdsc [1];
4         IF NOT nma$nparse (mom$ab_nparse_blk, mom$npa_mopinit) THEN
5             BEGIN
5             mom$error (nma$c_sts_lpr);
5             RETURN;
4             END;
4         !
4         ! Rewind the output file and reset the memory address for
4         ! retrying
4         !
4         mom$srvrewind ();
4         recidx = 0;
4         curmemadr = .memadr;
4         retries = .retries - 1;
3         END;
2     END;
2 !
2 ! Send a Dump Complete MOP message.
2 !
2 status = mom$mopsend (mom$ab_cib,
2                       1,
2                       UPLIT BYTE (mop$_fct_dcm));
2 !
2 ! Close the dump file.
2 !
2 mom$srvclose ();
2 !
2 ! If auto-dumping, tell the remote system to re-load itself by sending an
2 ! 'Enter MOP Mode' message.
2 !
2 IF .mom$gl_service_flags [mom$v_autoservice] THEN
2     mom$trigger ();
2 !
2 ! Return status.
2 !
2 mom$ab_msgblock [msb$l_flags] = 0;
2 IF .retries NEQ 0 THEN
2     mom$ab_msgblock [msb$b_code]  = nma$c_sts_suc
2 ELSE
2     mom$ab_msgblock [msb$b_code]  = nma$c_sts_lpr;
2 mom$bld_reply (mom$ab_msgblock, msgsize);
2 $signal_msg (mom$ab_nice_xmit_buf, .msgsize);
1 END;                            ! End of mom$dump
```

```
                                                  .PSECT  $PLIT$,NOWRT,NOEXE,2

        67  6E  69  70  6D  75  44  00040 P.AAF:  .ASCII  \Dumping\                                        ;
                                    00047         .BLKB   1
                          00000007  00048 P.AAE:  .LONG   7                                                ;
                          00000000' 0004C         .ADDRESS P.AAF                                           ;
                                 01 00050 P.AAG:  .BYTE   1                                                ;


                                                  .PSECT  $CODE$,NOWRT,2

                               0FFC 00000         .ENTRY  MOM$DUMP, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-; 0576
                                                          R11                                          ;
            5E        FCB4  CE   9E 00002         MOVAB   -844(SP), SP                                 ;
 00000000G  EF              03   90 00007         MOVB    #3, MOM$GW_EVT_CODE                          ; 0637
 00000000G  EF              01   90 0000E         MOVB    #1, MOM$GB_EVT_PSER                          ; 0638
                            7E   7C 00015         CLRQ    -(SP)                                        ; 0639
 00000000V  EF              02   FB 00017         CALLS   #2, MOM$LOG_EVENT                            ;
                 00000000G  EF   DD 0001E         PUSHL   MOM$AB_CIB                                   ; 0642
                            08   DD 00024         PUSHL   #8                                           ; 0641
 00000000G  EF              02   FB 00026         CALLS   #2, MOM$MOPSETSUBSTATE                       ;
                        FC  AD   D4 0002D         CLRL    FILDSC+4                                     ; 0646
            50 00000000*    EF   9A 00030         MOVZBL  <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_DUM*137>-; 0647
                                                          >+8>, R0                                     ;
                            06   14 00037         BGTR    1$                                           ;
            7E          82  8F   9A 00039         MOVZBL  #130, -(SP)                                  ; 0649
                            19   11 0003D         BRB     2$                                           ;
       F8   AD              50   D0 0003F 1$:     MOVL    R0, FILDSC                                   ; 0654
       FC   AD 00000000*    EF   9E 00043         MOVAB   <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_DUM*137>-; 0655
                                                          >+9>, FILDSC+4                               ;
            6E 00000000*    EF   D0 0004B         MOVL    <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_DCT*137>-; 0660
                                                          >+9>, MEMSIZ                                 ;
                            0F   18 00052         BGEQ    3$                                           ; 0661
            7E          88  8F   9A 00054         MOVZBL  #136, -(SP)                                  ; 0663
            7E              1D   CE 00058 2$:     MNEGL   #29, -(SP)                                   ;
 00000000G  EF              02   FB 0005B         CALLS   #2, MOM$ERROR                                ;
                                 04 00062         RET                                                  ; 0662
            58 00000000*    EF   D0 00063 3$:     MOVL    <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_DAD*137>-; 0670
                                                          >+9>, MEMADR                                 ;
                            02   18 0006A         BGEQ    4$                                           ; 0671
                            58   D4 0006C         CLRL    MEMADR                                       ; 0672
                            01   DD 0006E 4$:     PUSHL   #1                                           ; 0677
                        F8  AD   9F 00070         PUSHAB  FILDSC                                       ;
 00000000G  EF              02   FB 00073         CALLS   #2, MOM$SRVOPEN                              ;
            5A              50   D0 0007A         MOVL    R0, STATUS                                   ;
            0A              5A   E8 0007D         BLBS    STATUS, 5$                                   ; 0679
 00000000G  EF              02   B0 00080         MOVW    #2, MOM$AB_MSGBLOCK+8                        ; 0681
                          0183   31 00087         BRW     16$                                          ; 0682
                 00000000'  EF   9F 0008A 5$:     PUSHAB  P.AAE                                        ; 0689
                            06   DD 00090         PUSHL   #6                                           ;
 00000000G  EF              02   FB 00092         CALLS   #2, MOM$DEBUG_TXT                            ;
       F0   AD              04   90 00099         MOVB    #4, RMDBUF                                   ; 0691
       F5   AD        0100  8F   B0 0009D         MOVW    #256, RMDBUF+5                               ; 0692
       E8   AD              07   D0 000A3         MOVL    #7, RMDDSC                                   ; 0693
       EC   AD          F0  AD   9E 000A7         MOVAB   RMDBUF, RMDDSC+4                             ; 0694
```

```
                                 FED8 CD       0105  8F  3C 000AC           MOVZWL   #261, MDDDSC                                 ; 0696
                                 FEDC CD       FEE0  CD  9E 000B3           MOVAB    MDDBUF, MDDDSC+4                             ; 0697
                                      5B  00000000G  EF  9E 000BA           MOVAB    MOM$GQ_MOP_MSG_DSC, SKIP_MSG_DSC_ADDR        ; 0701
                                      57             58  D0 000C1           MOVL     MEMADR, CURMEMADR                            ; 0702
                                                     56  D4 000C4           CLRL     RECIDX                                       ; 0703
                                      59             02  D0 000C6           MOVL     #2, RETRIES                                  ; 0704
                                      6E             57  D1 000C9 6$:       CMPL     CURMEMADR, MEMSIZ                            ; 0706
                                                     03  1F 000CC           BLSSU    8$                                           ;
                                                   00F5  31 000CE 7$:       BRW      13$                                          ;
                                                     59  D5 000D1 8$:       TSTL     RETRIES                                      ; 0707
                                                     F9  13 000D3           BEQL     7$                                           ;
                                   F1 AD             57  D0 000D5           MOVL     CURMEMADR, RMDBUF+1                          ; 0715
                                                     5B  DD 000D9           PUSHL    SKIP_MSG_DSC_ADDR                            ; 0719
                                                 08  AE  9F 000DB           PUSHAB   MDDLEN                                       ; 0716
                                               FED8  CD  9F 000DE           PUSHAB   MDDDSC                                       ;
                                          00000000G  EF  9F 000E2           PUSHAB   MOM$AB_CIB                                   ;
                                                 E8  AD  9F 000E8           PUSHAB   RMDDSC                                       ;
                                          00000000G  EF  9F 000EB           PUSHAB   MOM$AB_CIB                                   ;
                            00000000G EF             06  FB 000F1           CALLS    #6, MOM$MOPSNDRCV                            ;
                                      5A             50  D0 000F8           MOVL     R0, STATUS                                   ;
                                                     5B  D4 000FB           CLRL     SKIP_MSG_DSC_ADDR                            ; 0720
                                      4D             5A  E9 000FD           BLBC     STATUS, 10$                                  ; 0725
                                      05         04  AE  D1 00100           CMPL     MDDLEN, #5                                   ; 0726
                                                     47  1F 00104           BLSSU    10$                                          ;
                                      0E       FEE0  CD  91 00106           CMPB     MDDBUF, #14                                  ; 0727
                                                     40  12 0010B           BNEQ     10$                                          ;
                                      57       FEE1  CD  D1 0010D           CMPL     MDDBUF+1, CURMEMADR                          ; 0728
                                                     39  12 00112           BNEQ     10$                                          ;
                          50       04 AE             05  C3 00114           SUBL3    #5, MDDLEN, R0                               ; 0730
                    24 AE46      FEE5 CD             50  28 00119           MOVC3    R0, MDDBUF+5, RECBUF[RECIDX]                 ; 0731
           7E 00000100   8F           56             01  7A 00121           EMUL     #1, RECIDX, #256, -(SP)                      ; 0732
           56            56           8E   00000200  8F  7B 0012A           EDIV     #512, (SP)+, RECIDX, RECIDX                  ;
                                                     56  D5 00133           TSTL     RECIDX                                       ; 0737
                                                     0F  12 00135           BNEQ     9$                                           ;
                                      7E       0200  8F  3C 00137           MOVZWL   #512, -(SP)                                  ; 0738
                                                 28  AE  9F 0013C           PUSHAB   RECBUF                                       ;
                            00000000G EF             02  FB 0013F           CALLS    #2, MOM$SRVWRITE                             ;
                                      57       0100  C7  9E 00146 9$:       MOVAB    256(R7), CURMEMADR                           ; 0740
                                                     76  11 0014B           BRB      12$                                          ; 0725
                                                 14  AE  9F 0014D 10$:      PUSHAB   SNDDSC                                       ; 0749
                            00000000G EF             01  FB 00150           CALLS    #1, MOM$BLDMOPRDS                            ;
                                   10 AE  00000000G  EF  D0 00157           MOVL     MOM$GQ_MOP_RCV_BUF_DSC+4, MSGDSC+4           ; 0750
                                                     7E  D4 0015F           CLRL     -(SP)                                        ; 0752
                                                 10  AE  9F 00161           PUSHAB   MSGDSC                                       ; 0754
                                          00000000G  EF  9F 00164           PUSHAB   MOM$GQ_MOP_RCV_BUF_DSC                       ; 0752
                                          00000000G  EF  9F 0016A           PUSHAB   MOM$AB_CIB                                   ;
                                                 24  AE  9F 00170           PUSHAB   SNDDSC                                       ;
                                          00000000G  EF  9F 00173           PUSHAB   MOM$AB_CIB                                   ;
                            00000000G EF             06  FB 00179           CALLS    #6, MOM$MOPSNDRCV                            ;
                                      5A             50  D0 00180           MOVL     R0, STATUS                                   ;
                                                     5A  DD 00183           PUSHL    STATUS                                       ; 0756
                            00000000V EF             01  FB 00185           CALLS    #1, MOM$CHK_MOP_ERROR                        ;
                            00000000G EF         0C  AE  7D 0018C           MOVQ     MSGDSC, MOM$AB_NPARSE_BLK+4                  ; 0757
                                          00000000G  EF  9F 00194           PUSHAB   MOM$NPA_MOPINIT                              ; 0759
                                          00000000G  EF  9F 0019A           PUSHAB   MOM$AB_NPARSE_BLK                            ;
                            00000000G EF             02  FB 001A0           CALLS    #2, NMA$NPARSE                               ;
                                      0B             50  E8 001A7           BLBS     R0, 11$                                      ;
```

```
                     7E            11  CE 001AA          MNEGL   #17, -(SP)                                    ; 0761
           00000000G EF            01  FB 001AD          CALLS   #1, MOM$ERROR                                 ;
                                       04 001B4          RET                                                   ; 0760
           00000000G EF            00  FB 001B5 11$:     CALLS   #0, MOM$SRVREWIND                             ; 0768
                                   56  D4 001BC          CLRL    RECIDX                                        ; 0769
                     57            58  D0 001BE          MOVL    MEMADR, CURMEMADR                             ; 0770
                                   59  D7 001C1          DECL    RETRIES                                       ; 0771
                                 FF03  31 001C3 12$:     BRW     6$                                            ; 0706
                      00000000'    EF  9F 001C6 13$:     PUSHAB  P.AAG                                         ; 0779
                                   01  DD 001CC          PUSHL   #1                                            ; 0777
                      00000000G    EF  9F 001CE          PUSHAB  MOM$AB_CIB                                    ;
           00000000G EF            03  FB 001D4          CALLS   #3, MOM$MOPSEND                               ;
                     5A            50  D0 001DB          MOVL    R0, STATUS                                    ;
           00000000G EF            00  FB 001DE          CALLS   #0, MOM$SRVCLOSE                              ; 0783
                     07 00000000G  EF  E9 001E5          BLBC    MOM$GL_SERVICE_FLAGS, 14$                     ; 0788
           00000000V EF            00  FB 001EC          CALLS   #0, MOM$TRIGGER                               ; 0789
                      00000000G    EF  D4 001F3 14$:     CLRL    MOM$AB_MSGBLOCK                               ; 0793
                                   59  D5 001F9          TSTL    RETRIES                                       ; 0794
                                   09  13 001FB          BEQL    15$                                           ;
           00000000G EF            01  90 001FD          MOVB    #1, MOM$AB_MSGBLOCK+4                         ; 0795
                                   07  11 00204          BRB     16$                                           ;
           00000000G EF            11  8E 00206 15$:     MNEGB   #17, MOM$AB_MSGBLOCK+4                        ; 0797
                             08    AE  9F 0020D 16$:     PUSHAB  MSGSIZE                                       ; 0798
                      00000000G    EF  9F 00210          PUSHAB  MOM$AB_MSGBLOCK                               ;
           00000000G EF            02  FB 00216          CALLS   #2, MOM$BLD_REPLY                             ;
                             08    AE  DD 0021D          PUSHL   MSGSIZE                                       ; 0799
                      00000000G    EF  9F 00220          PUSHAB  MOM$AB_NICE_XMIT_BUF                          ;
                      02070000     8F  DD 00226          PUSHL   #34013T84                                     ;
           00000000G 00            03  FB 0022C          CALLS   #3, LIB$SIGNAL                                ;
                                   04  00233             RET                                                   ; 0800

; Routine Size:  564 bytes,    Routine Base:  $CODE$ + 032D
```

```
1 %SBTTL 'mom$trigger Trigger the target bootstrap'
1 GLOBAL ROUTINE mom$trigger =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !           This routine performs the trigger bootstrap function.  It is called
1 !           during trigger, and dump operations.
1 !
1 ! INPUTS:
1 !           None
1 !
1 ! IMPLICIT INPUTS:
1 !           MOM$AB_CIB = Channel Information for QIO channel for MOP messages.
1 !           MOM$GB_FUNCTION = maintenance function currently being performed.
1 !
1 ! ROUTINE VALUE:
1 ! COMPLETION CODES:
1 !
1 !           Signal errors.
1 !
1 !--
1
2 BEGIN
2
2 LOCAL
2     snddsc :    VECTOR [2],
2     status;
2
2 !
2 ! Open an I/O channel for the circuit service operation.
2 !
2 mom$mopopen (mom$ab_trigger_cib [cib$l_chan]);
2 !
2 ! Set up the Channel Information Block for the channel.  For NI circuits, this
2 ! sets up the NI protocol for the circuit, and determines the NI destination
2 ! address that the target DEUNA is currently responding to.
2 !
2 mom$init_CIB (mom$ab_trigger_cib,
2                 nma$c_fnc_tri,
2                 svd$gk_pcno_pha,
2                 svd$gk_pcno_add,
2                 svd$gk_pcno_hwa);
2 IF .mom$gb_function EQL nma$c_fnc_tri THEN
3     BEGIN
3     !
3     ! Set the circuit substate to -TRIGGERING.
3     !
3     mom$mopsetsubstate (nma$c_linss_tri,
3                         .mom$ab_trigger_cib [cib$l_chan]);
2     END;
2 !
2 ! Build the trigger (old 'enter MOP mode', new 'boot') message.
2 !
2 mom$bldmopboot (snddsc);
2 mom$debug_txt (dbg$c_srvtrc,
2                 $ASCID ('Triggering remote bootstrap'));
2 !
```

```
;   865            0858  2  ! Send the trigger message for a trigger or dump operation, and don't
;   866            0859  2  ! wait for a response message from the target.
;   867            0860  2  !
;   868            0861  2  status = mom$mopsend (mom$ab_trigger_cib,
;   869            0862  2                        .snddsc [1],
;   870            0863  2                        .snddsc [0]);
;   871            0864  2  !
;   872            0865  2  ! If the target's correct physical address is not available, send the
;   873            0866  2  ! trigger to the target's hardware address as well.
;   874            0867  2  !
;   875            0868  2  IF NOT .mom$ab_trigger_cib [cib$v_target_addr_fixed] THEN
;   876            0869  3      BEGIN
;   877            0870  3      mom$set_ni_addr (mom$ab_trigger_cib);
;   878            0871  3      status = mom$mopsend (mom$ab_trigger_cib,
;   879            0872  3                            .snddsc [1],
;   880            0873  3                            .snddsc [0]);
;   881            0874  2      END;
;   882            0875  2  mom$chk_mop_error (.status);
;   883            0876  2  !
;   884            0877  2  ! If it's a trigger operation, return a NICE response to NCP (trigger is
;   885            0878  2  ! never an autoservice operation), and quit.
;   886            0879  2  !
;   887            0880  2  IF .mom$gb_function EQL nma$c_fnc_tri THEN
;   888            0881  2      mom$error (nma$c_sts_suc);
;   889            0882  2  RETURN .status;
;   890            0883  1  END;                                 ! End of mom$trigger



                                                                     .PSECT  $PLIT$,NOWRT,NOEXE,2

6F  6D  65  72  20  67  6E  69  72  65  67  67  69  72  54  00051 P.AAI:  .ASCII  \Triggering remote bootstrap\                ;
            70  61  72  74  73  74  6F  6F  62  20  65  74  00060                                                            ;
                                                0000001B  0006C P.AAH:  .LONG   27                                          ;
                                                00000000' 00070         .ADDRESS P.AAI                                      ;



                                                                     .PSECT  $CODE$,NOWRT,2

                                                     003C 00000         .ENTRY  MOM$TRIGGER, Save R2,R3,R4,R5              ; 0802
                           55 00000000G   EF      9E 00002         MOVAB   MOM$MOPSEND, R5                            ;
                           54 00000000G   EF      9E 00009         MOVAB   MOM$GB_FUNCTION, R4                        ;
                           53 00000000G   EF      9E 00010         MOVAB   MOM$AB_TRIGGER_CIB, R3                     ;
                           5E                  08 C2 00017         SUBL2   #8, SP                                     ;
                                               53 DD 0001A         PUSHL   R3                                         ; 0832
                    00000000G  EF              01 FB 0001C         CALLS   #1, MOM$MOPOPEN                            ;
                               00000000G  8F   DD 00023         PUSHL   #SVD$GK_PCNO_HWA                           ; 0838
                               00000000G  8F   DD 00029         PUSHL   #SVD$GK_PCNO_ADD                           ;
                               00000000G  8F   DD 0002F         PUSHL   #SVD$GK_PCNO_PHA                           ;
                                               11 DD 00035         PUSHL   #17                                        ;
                                               53 DD 00037         PUSHL   R3                                         ;
                    00000000G  EF              05 FB 00039         CALLS   #5, MOM$INIT_CIB                           ;
                           11                  64 91 00040         CMPB    MOM$GB_FUNCTION, #17                       ; 0843
                                               0B 12 00043         BNEQ    1$                                         ;
                                               63 DD 00045         PUSHL   MOM$AB_TRIGGER_CIB                         ; 0849
                                               05 DD 00047         PUSHL   #5                                         ; 0848
```

```
           00000000G  EF            02  FB 00049      CALLS   #2, MOM$MOPSETSUBSTATE                      ;
                                    5E  DD 00050 1$:  PUSHL   SP                                          ; 0854
           00000000G  EF            01  FB 00052      CALLS   #1, MOM$BLDMOPBOOT                          ;
                      00000000'     EF  9F 00059      PUSHAB  P.AAH                                       ; 0856
                                    06  DD 0005F      PUSHL   #6                                          ; 0855
           00000000G  EF            02  FB 00061      CALLS   #2, MOM$DEBUG_TXT                           ;
                                    6E  DD 00068      PUSHL   SNDDSC                                      ; 0863
                            08      AE  DD 0006A      PUSHL   SNDDSC+4                                    ; 0862
                                    53  DD 0006D      PUSHL   R3                                          ; 0861
                      65            03  FB 0006F      CALLS   #3, MOM$MOPSEND                             ;
                      52            50  D0 00072      MOVL    R0, STATUS                                  ;
                      16    10      A3  E8 00075      BLBS    MOM$AB_TRIGGER_CIB+16, 2$                   ; 0868
                                    53  DD 00079      PUSHL   R3                                          ; 0870
           00000000G  EF            01  FB 0007B      CALLS   #1, MOM$SET_NI_ADDR                         ;
                                    6E  DD 00082      PUSHL   SNDDSC                                      ; 0873
                            08      AE  DD 00084      PUSHL   SNDDSC+4                                    ; 0872
                                    53  DD 00087      PUSHL   R3                                          ; 0871
                      65            03  FB 00089      CALLS   #3, MOM$MOPSEND                             ;
                      52            50  D0 0008C      MOVL    R0, STATUS                                  ;
                                    52  DD 0008F 2$:  PUSHL   STATUS                                      ; 0875
           00000000V  EF            01  FB 00091      CALLS   #1, MOM$CHK_MOP_ERROR                       ;
                      11            64  91 00098      CMPB    MOM$GB_FUNCTION, #17                        ; 0880
                                    09  12 0009B      BNEQ    3$                                          ;
                                    01  DD 0009D      PUSHL   #1                                          ; 0881
           00000000G  EF            01  FB 0009F      CALLS   #1, MOM$ERROR                               ;
                      50            52  D0 000A6 3$:  MOVL    STATUS, R0                                  ; 0882
                                        04 000A9      RET                                                 ; 0883

; Routine Size:  170 bytes,    Routine Base:  $CODE$ + 0561
;  892      0884  1 %SBTTL 'mom$servicehandler  Condition handler'
;  893      0885  1 GLOBAL ROUTINE mom$servicehandler (signal_vec, mechanism)       =
;  894      0886  1
;  895      0887  1 !++
;  896      0888  1 ! FUNCTIONAL DESCRIPTION:
;  897      0889  1 !
;  898      0890  1 !       This routine is a condition handler that performs cleanup
;  899      0891  1 !       at the end of maintenance operations.  The MOP I/O channel is
;  900      0892  1 !       deassigned.  T
;  901      0893  1 !
;  902      0894  1 ! FORMAL PARAMETERS:
;  903      0895  1 !
;  904      0896  1 !       SIGNAL_VEC        Pointer to the signal vector.
;  905      0897  1 !       MECHANISM         Pointer to the mechanism array.
;  906      0898  1 !
;  907      0899  1 !--
;  908      0900  2 BEGIN
;  909      0901  2
;  910      0902  2 MAP
;  911      0903  2     signal_vec : REF BBLOCK,               ! Signal vector argument
;  912      0904  2     mechanism  : REF BBLOCK;               ! Mechanism vector array pointer
;  913      0905  2
;  914      0906  2 LOCAL
;  915      0907  2     buf_adr,                               ! Temporary buffer address
;  916      0908  2     buf_len,                               ! Temporary buffer length
;  917      0909  2     sts_code : BBLOCK [4];                 ! Status code
;  918      0910  2
;  919      0911  2 !
;  920      0912  2 ! Deassign the MOP channels.
;  921      0913  2 !
;  922      0914  2 IF .mom$ab_cib [cib$l_chan] NEQ 0 THEN
;  923      0915  2     $DASSGN (CHAN = .mom$ab_cib [cib$l_chan]);
;  924      0916  2 IF .mom$ab_trigger_cib [cib$l_chan] NEQ 0 THEN
;  925      0917  2     $DASSGN (CHAN = .mom$ab_trigger_cib [cib$l_chan]);
;  926      0918  2
;  927      0919  2 sts_code = .signal_vec [chf$l_sig_name]; ! Get signal status code
;  928      0920  2 !
;  929      0921  2 ! If the facility code matches the one for MOM, then MOM signalled the
;  930      0922  2 ! error with it's own signalling arguments.  Use these arguments to
;  931      0923  2 ! send the response message back to NML.
;  932      0924  2 !
;  933      0925  2 IF .sts_code [sts$v_fac_no] EQLU mom$k_fac_code THEN
;  934      0926  3     BEGIN
;  935      0927  3     !
;  936      0928  3     ! Two arguments are required for MOM conditions.
;  937      0929  3     !
;  938      0930  3     IF .signal_vec [chf$l_sig_args] NEQU 2+3 THEN
;  939      0931  3         RETURN ss$_resignal
;  940      0932  3     ELSE
;  941      0933  4         BEGIN
;  942      0934  4
;  943      0935  4         buf_adr = .signal_vec [chf$l_sig_arg1];
;  944      0936  4         buf_len = .(signal_vec [chf$l_sig_arg1]+4);
;  945      0937  4         !
;  946      0938  4         ! For load operations, log the final status event.
;  947      0939  4         !
;  948      0940  4         IF .mom$gb_function EQL nma$c_fnc_loa THEN
;  949      0941  4                 mom$log_event (.buf_len, .buf_adr);
;  950      0942  4                 !
;  951      0943  4                 ! If a message is specified (length not equal 0) then send it.
;  952      0944  4                 !
;  953      0945  4                 IF .buf_len NEQU 0 THEN
;  954      0946  4                     mom$send (.buf_adr, .buf_len); ! Send status message
;  955      0947  4
;  956      0948  4                 !
;  957      0949  4                 ! Unwind back to the caller of the routine that set up the
;  958      0950  4                 ! condition hanlder and continue from there.  This gets rid of
;  959      0951  4                 ! the MOM specific arguments on the stack.
;  960      0952  4                 !
;  961      0953  4                 $UNWIND ();
;  962      0954  4                 RETURN ss$_continue
;  963      0955  4
;  964      0956  3                 END;
;  965      0957  3             END
;  966      0958  2         ELSE
;  967      0959  2             !
;  968      0960  2             ! This condition was not signalled by MOM so let it go by.
;  969      0961  2             !
;  970      0962  2             RETURN ss$_resignal
;  971      0963  2
;  972      0964  1         END;                                    ! End of MOM$SERVICEHANDLER


                                                                     .EXTRN  SYS$DASSGN, SYS$UNWIND

                                                    001C 00000       .ENTRY  MOM$SERVICEHANDLER, Save R2,R3,R4                ; 0885
                               54 00000000G   00      9E 00002       MOVAB   SYS$DASSGN, R4                                   ;
                               50 00000000G   EF      D0 00009       MOVL    MOM$AB_CIB, R0                                   ; 0914
                                              05      13 00010       BEQL    1$                                               ;
                                              50      DD 00012       PUSHL   R0                                               ; 0915
                               64             01      FB 00014       CALLS   #1, SYS$DASSGN                                   ;
                               50 00000000G   EF      D0 00017 1$:   MOVL    MOM$AB_TRIGGER_CIB, R0                           ; 0916
                                              05      13 0001E       BEQL    2$                                               ;
                                              50      DD 00020       PUSHL   R0                                               ; 0917
                               64             01      FB 00022       CALLS   #1, SYS$DASSGN                                   ;
                               50         04  AC      D0 00025 2$:   MOVL    SIGNAL_VEC, R0                                   ; 0919
                               51         04  A0      D0 00029       MOVL    4(R0), STS_CODE                                  ;
00000207   8F           51     0C             10      ED 0002D       CMPZV   #16, #12, STS_CODE, #519                         ; 0925
                                              3B      12 00036       BNEQ    5$                                               ;
                               05             60      D1 00038       CMPL    (R0), #5                                         ; 0930
                                              36      12 0003B       BNEQ    5$                                               ;
                               53         08  A0      D0 0003D       MOVL    8(R0), BUF_ADR                                   ; 0935
                               52         0C  A0      D0 00041       MOVL    12(R0), BUF_LEN                                  ; 0936
                               0F 00000000G   EF      91 00045       CMPB    MOM$GB_FUNCTION, #15                             ; 0940
                                              09      12 0004C       BNEQ    3$                                               ;
                                              0C      BB 0004E       PUSHR   #^M<R2,R3>                                       ; 0941
                   00000000V   EF             02      FB 00050       CALLS   #2, MOM$LOG_EVENT                                ;
                                              52      D5 00057 3$:   TSTL    BUF_LEN                                          ; 0945
                                              0B      13 00059       BEQL    4$                                               ;
                                              52      DD 0005B       PUSHL   BUF_LEN                                          ; 0946
                                              53      DD 0005D       PUSHL   BUF_ADR                                          ;
                   00000000G   EF             02      FB 0005F       CALLS   #2, MOM$SEND                                     ;
                                              7E      7C 00066 4$:   CLRQ    -(SP)                                            ; 0953
```

```
                          00000000G  00              02  FB 00068        CALLS   #2, SYS$UNWIND                                       ;
                                     50              01  D0 0006F        MOVL    #1, R0                                               ; 0954
                                                         04 00072        RET                                                          ; 0962
                                     50      0918    8F  3C 00073 5$:    MOVZWL  #2328, R0                                            ;
                                                         04 00078        RET                                                          ; 0964

; Routine Size:  121 bytes,    Routine Base:  $CODE$ + 060B
```

```
1 %SBTTL 'mom$autohandler  Condition handler for autoservice operations'
1 GLOBAL ROUTINE mom$autohandler (SIGNAL_VEC, MECHANISM)  =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         This routine is a condition handler that performs cleanup
1 !         at the end of maintenance operations initiated by the target
1 !         (autoservice).  The final status is logged and the direct I/O
1 !         channel is deassigned.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         SIGNAL_VEC          Pointer to the signal vector.
1 !         MECHANISM           Pointer to the mechanism array.
1 !
1 !--
2 BEGIN
2
2 MAP
2     signal_vec : REF BBLOCK,      ! Signal vector argument
2     mechanism  : REF BBLOCK;      ! Mechanism vector array pointer
2
2 LOCAL
2     msgadr,
2     msgsize,
2     sts_code : BBLOCK [4];                  ! Status code
2
2 sts_code = .signal_vec [chf$l_sig_name];
2
2 !
2 ! If the error was generated by MOM (as opposed to being an SS$_ error)
2 ! log an event.
2 !
2 IF .sts_code [sts$v_fac_no] EQLU mom$k_fac_code THEN
3     BEGIN
3
3     msgadr = .signal_vec [chf$l_sig_arg1];
3     msgsize = .(signal_vec [chf$l_sig_arg1]+4);
3     !
3     ! Log the event.
3     !
3     mom$log_event (.msgsize, .msgadr);
3     !
3     ! Unwind back to the caller of the routine that set up the
3     ! condition hanlder and continue from there.  This gets rid of
3     ! the MOM specific arguments on the stack.
3     !
3     $UNWIND ();
3     RETURN ss$_continue
3     END;
2 !
2 ! Deassign the MOP channel.
2 !
2 IF .mom$ab_cib [cib$l_chan] NEQ 0 THEN
2     $DASSGN (CHAN = .mom$ab_cib [cib$l_chan]);
2
```

```
; 1031           1022  2 RETURN ss$_resignal;                   ! Always resignal error
; 1032           1023  2
; 1033           1024  1 END;                                   ! End of MOM$AUTOHANDLER



                                                  0000 00000            .ENTRY   MOM$AUTOHANDLER, Save nothing                  ; 0966
                                  50        04   AC  D0 00002            MOVL     SIGNAL_VEC, R0                                 ; 0993
                                  51        04   A0  D0 00006            MOVL     4(R0), STS_CODE
00000207   8F              51     0C             10  ED 0000A            CMPZV    #16, #12, STS_CODE, #519                       ; 0999
                                                 1E  12 00013            BNEQ     1$
                                  51        08   A0  D0 00015            MOVL     8(R0), MSGADR                                  ; 1002
                                  50        0C   A0  D0 00019            MOVL     12(R0), MSGSIZE                                ; 1003
                                                 03  BB 0001D            PUSHR    #^M<R0,R1>                                     ; 1007
                       00000000V  EF             02  FB 0001F            CALLS    #2, MOM$LOG_EVENT
                                                 7E  7C 00026            CLRQ     -(SP)                                          ; 1013
                       00000000G  00             02  FB 00028            CALLS    #2, SYS$UNWIND
                                  50             01  D0 0002F            MOVL     #1, R0                                         ; 1014
                                                     04 00032            RET
                                  50 00000000G   EF  D0 00033 1$:        MOVL     MOM$AB_CIB, R0                                 ; 1019
                                                 09  13 0003A            BEQL     2$
                                                 50  DD 0003C            PUSHL    R0                                             ; 1020
                       00000000G  00             01  FB 0003E            CALLS    #1, SYS$DASSGN
                                  50      0918   8F  3C 00045 2$:        MOVZWL   #2328, R0                                      ; 1022
                                                     04 0004A            RET                                                     ; 1024

; Routine Size:  75 bytes,    Routine Base:  $CODE$ + 0684
```

```
1 %SBTTL 'mom$log_event  Log the network management layer event'
1 GLOBAL ROUTINE mom$log_event (msgsize, msgadr) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Log the network management event.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       MSGSIZE         Size of response message (if necessary).
1 !       MSGADR          Address of response message (if necessary).
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       MOM$GW_EVT_CODE
1 !       MOM$GB_EVT_PSER
1 !       MOM$GB_EVT_POPR
1 !       MOM$GB_EVT_PRSN
1 !
1 !--
2 BEGIN
2
2 LOCAL
2     status,
2     evtbuf : BBLOCK [mom$k_qio_buf_len],
2     evtdsc : VECTOR [2],
2     nfb    : BBLOCK [nfb$k_length],
2     nfbdsc : VECTOR [2],
2     svd_index,
2     ptr,
2     node_cmu;           ! Coded multiple field count for target node ID
2
2 !
2 ! Initialize the event buffer pointer leaving room for the byte count word.
2 !
2 ptr = evtbuf + 2;
2 !
2 ! Get the current system time and put it in the buffer.
2 !
2 $GETTIM (timadr = .ptr);
2 ptr = .ptr + 8;
2 !
2 ! Put the event code in the buffer.
2 !
2 (.ptr)<0,16> = .mom$gw_evt_code;
2 ptr = .ptr + 2;
2 !
2 ! Add the source circuit id.
2 !
2 CH$WCHAR_A (evc$c_src_cir, ptr);        ! Add source type code (Always CIRCUIT)
2 CH$WCHAR_A (.mom$ab_service_data [svd$gk_pcno_sli, svd$b_string_len], ptr);
2 ptr = CH$COPY (.mom$ab_service_data [svd$gk_pcno_sli, svd$b_string_len],
2               mom$ab_service_data [svd$gk_pcno_sli, svd$t_string],
2               0,
2               16,
2               .ptr);
```

```
; 1092      1082  2  !
; 1093      1083  2  ! Add the event-specific information.
; 1094      1084  2  !
; 1095      1085  2  SELECTU .mom$gw_evt_code OF
; 1096      1086  2      SET
; 1097      1087  2      !
; 1098      1088  2      ! Add service code.
; 1099      1089  2      !
; 1100      1090  2      [evc$c_nma_als]:
; 1101      1091  3          BEGIN
; 1102      1092  3          (.ptr)<0,16> = evc$c_nma_pser;
; 1103      1093  3          ptr = .ptr + 2;
; 1104      1094  3          CH$WCHAR_A (nma$m_pty_cod or 1, ptr);
; 1105      1095  3          CH$WCHAR_A (.mom$gb_evt_pser, ptr);
; 1106      1096  2          END;
; 1107      1097  2      !
; 1108      1098  2      ! Add operation code.
; 1109      1099  2      !
; 1110      1100  2      [evc$c_nma_psl]:
; 1111      1101  3          BEGIN
; 1112      1102  3          (.ptr)<0,16> = evc$c_nma_popr;
; 1113      1103  3          ptr = .ptr + 2;
; 1114      1104  3          CH$WCHAR_A (nma$m_pty_cod or 1, ptr);
; 1115      1105  3          CH$WCHAR_A (.mom$gb_evt_popr, ptr);
; 1116      1106  2          END;
; 1117      1107  2      !
; 1118      1108  2      ! Add reason code.
; 1119      1109  2      !
; 1120      1110  2      [evc$c_nma_abs]:
; 1121      1111  3          BEGIN
; 1122      1112  3          (.ptr)<0,16> = evc$c_nma_prsn;
; 1123      1113  3          ptr = .ptr + 2;
; 1124      1114  3          CH$WCHAR_A (nma$m_pty_cod OR 1, ptr);
; 1125      1115  3          CH$WCHAR_A (.mom$gb_evt_prsn, ptr);
; 1126      1116  2          END;
; 1127      1117  2      !
; 1128      1118  2      ! Add status message.
; 1129      1119  2      !
; 1130      1120  2      [evc$c_nma_abs,evc$c_nma_als]:
; 1131      1121  3          BEGIN
; 1132      1122  3          (.ptr)<0,16> = evc$c_nma_psts;
; 1133      1123  3          ptr = .ptr + 2;
; 1134      1124  3          CH$WCHAR_A (nma$m_pty_cmu OR 3, ptr);
; 1135      1125  3
; 1136      1126  3          CH$WCHAR_A (nma$m_pty_cod OR 1, ptr);
; 1137      1127  3          IF .msgsize GTR 0 THEN
; 1138      1128  3              CH$WCHAR_A (.(.msgadr)<0,8>, ptr)
; 1139      1129  3          ELSE
; 1140      1130  3              CH$WCHAR_A (0, ptr);
; 1141      1131  3
; 1142      1132  3          CH$WCHAR_A (nma$m_pty_cod OR 2, ptr);
; 1143      1133  3          IF .msgsize GTR 2 THEN
; 1144      1134  3              (.ptr)<0,16> = .(.msgadr + 1)<0,16>
; 1145      1135  3          ELSE
; 1146      1136  3              (.ptr)<0,16> = -1;
; 1147      1137  3          ptr = .ptr + 2;
; 1148      1138  3
```

```
3            CH$WCHAR_A (nma$m_pty_asc, ptr);
3            IF .msgsize GTR 4 THEN
4                BEGIN
4
4                CH$WCHAR_A (.(.msgadr + 3)<0,8>, ptr);
4                ptr = CH$MOVE (.(.msgadr + 3)<0,8>,
4                               .msgadr + 4,
4                               .ptr);
4                END
3            ELSE
3                CH$WCHAR_A (0, ptr);
3
2            END;
2        !
2        ! If it's in the Service Data Table (SVD) add the target's node ID.
2        !
2        [evc$c_nma_abs, evc$c_nma_als]:
3            BEGIN
3            node_cmu = 0;
3            IF .mom$ab_service_data [svd$gk_pcno_add, svd$l_param] NEQ 0 THEN
3                node_cmu = 1;
3            IF .mom$ab_service_data [svd$gk_pcno_nna, svd$b_string_len] GTR 0 THEN
3                node_cmu = .node_cmu + 1;
3            IF .node_cmu GTR 0 THEN
4                BEGIN
4                (.ptr)<0,16> = evc$c_nma_pnod;
4                ptr = .ptr + 2;
4                CH$WCHAR_A (nma$m_pty_cmu OR .node_cmu, ptr);
4                IF .mom$ab_service_data [svd$gk_pcno_add, svd$l_param] NEQ 0 THEN
5                    BEGIN
5                    CH$WCHAR_A (2, ptr);
5                    (.ptr)<0,16> = .mom$ab_service_data [svd$gk_pcno_add,
5                                                         svd$l_param];
5                    ptr = .ptr + 2;
4                    END;
4                IF .mom$ab_service_data [svd$gk_pcno_nna, svd$b_string_len] GTR 0
4                THEN
5                    BEGIN
5                    CH$WCHAR_A (nma$m_pty_asc,
5                                ptr);
5                    CH$WCHAR_A (.mom$ab_service_data [svd$gk_pcno_nna, svd$b_string_len],
5                                ptr);
5                    ptr = CH$MOVE (.mom$ab_service_data [svd$gk_pcno_nna,
5                                                         svd$b_string_len],
5                                    mom$ab_service_data [svd$gk_pcno_nna,
5                                                         svd$t_string],
5
5                                    .ptr);
4                    END;
3                END;
2            END;
2        !
2        ! Add file and software type to load events.
2        !
2        [evc$c_nma_als]:
3            BEGIN
3            IF .mom$gb_evt_pser EQL evc$c_nma_pser_loa THEN
4                BEGIN
```

```
; 1206          1196  4              !
; 1207          1197  4              ! File ID of load file.
; 1208          1198  4              !
; 1209          1199  4              (.ptr)<0,16> = evc$c_nma_pfil;
; 1210          1200  4              ptr = .ptr + 2;
; 1211          1201  4              SELECTONE .mom$ab_service_data [svd$gk_pcno_sty, svd$l_param] OF
; 1212          1202  4                  SET
; 1213          1203  4                  [nma$c_soft_secl]: svd_index = svd$gk_pcno_slo;
; 1214          1204  4                  [nma$c_soft_terl]: svd_index = svd$gk_pcno_tlo;
; 1215          1205  4                  [nma$c_soft_osys]: svd_index = svd$gk_pcno_loa;
; 1216          1206  4                  TES;
; 1217          1207  4              CH$WCHAR_A (nma$m_pty_asc,
; 1218          1208  4                      ptr);
; 1219          1209  4              CH$WCHAR_A (.mom$ab_service_data [.svd_index, svd$b_string_len],
; 1220          1210  4                      ptr);
; 1221          1211  4              ptr = CH$MOVE (.mom$ab_service_data [.svd_index, svd$b_string_len],
; 1222          1212  4                      mom$ab_service_data [.svd_index, svd$t_string],
; 1223          1213  4                      .ptr);
; 1224          1214  4              !
; 1225          1215  4              ! Software type of load file.
; 1226          1216  4              !
; 1227          1217  4              (.ptr)<0,16> = evc$c_nma_psty;
; 1228          1218  4              ptr = .ptr + 2;
; 1229          1219  4              CH$WCHAR_A (nma$c_pty_cd1, ptr);
; 1230          1220  4              CH$WCHAR_A (.mom$ab_service_data [svd$gk_pcno_sty, svd$l_param],
; 1231          1221  4                          ptr);
; 1232          1222  3              END;
; 1233          1223  2          END;
; 1234          1224  2      !
; 1235          1225  2      ! If the circuit is an Ethernet, add the target system's Ethernet address
; 1236          1226  2      ! to the event.
; 1237          1227  2      !
; 1238          1228  2      [evc$c_nma_abs,evc$c_nma_als]:
; 1239          1229  3          BEGIN
; 1240          1230  3          IF .mom$gl_service_flags [mom$v_ni_circ] THEN
; 1241          1231  4              BEGIN
; 1242          1232  4              !
; 1243          1233  4              ! Add parameter ID = source NI address
; 1244          1234  4              !
; 1245          1235  4              (.ptr)<0,16> = evc$c_nma_psni;
; 1246          1236  4              ptr = .ptr + 2;
; 1247          1237  4              CH$WCHAR_A (nma$c_pty_hi,            ! It's a hex image field
; 1248          1238  4                          ptr);
; 1249          1239  4              CH$WCHAR_A (mom$k_ni_addr_length,
; 1250          1240  4                          ptr);
; 1251          1241  4              !
; 1252          1242  4              ! Add the Ethernet address.
; 1253          1243  4              !
; 1254          1244  4              ptr = CH$MOVE (mom$k_ni_addr_length,
; 1255          1245  4                      mom$ab_service_data [svd$gk_pcno_pha, svd$t_string],
; 1256          1246  4                      .ptr);
; 1257          1247  3              END;
; 1258          1248  2          END;
; 1259          1249  2      TES;
; 1260          1250  2  !
; 1261          1251  2  ! Set up the descriptor of the event buffer.
; 1262          1252  2  !
; 1263        1253  2 evtdsc [0] = .ptr - evtbuf;
; 1264        1254  2 (evtbuf)<0,16> = .evtdsc [0];
; 1265        1255  2 evtdsc [1] = evtbuf;
; 1266        1256  2 !
; 1267        1257  2 ! Set up the NFB.
; 1268        1258  2 !
; 1269        1259  2 CH$FILL (0, nfb$k_length, nfb);
; 1270        1260  2 nfb [nfb$b_fct] = nfb$c_logevent;
; 1271        1261  2 nfbdsc [0] = nfb$k_length;
; 1272        1262  2 nfbdsc [1] = nfb;
; 1273        1263  2 !
; 1274        1264  2 ! Log the event to NETACP.
; 1275        1265  2 !
; 1276        1266  2 mom$netacp_qio (nfbdsc, evtdsc, 0, 0);
; 1277        1267  2 !
; 1278        1268  2 ! Dump out the event message.
; 1279        1269  2 !
; 1280        1270  2 mom$debug_msg (dbg$c_events,
; 1281        1271  2                .evtdsc [1],
; 1282        1272  2                .evtdsc [0],
; 1283        1273  2                $ASCID ('Event logged'));
; 1284        1274  2
; 1285        1275  1 END;                                    ! End of MOM$LOG_EVENT


                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

  64 65 67 67 6F 6C 20 74 6E 65 76 45  00074 P.AAK:       .ASCII  \Event logged\                               ;
                            0000000C   00080 P.AAJ:       .LONG   12                                           ;
                            00000000'  00084              .ADDRESS P.AAK                                       ;

                                                          .EXTRN  SYS$GETTIM

                                                          .PSECT  $CODE$,NOWRT,2

                                 03FC  00000              .ENTRY  MOM$LOG_EVENT, Save R2,R3,R4,R5,R6,R7,R8,R9 ; 1026
               59 00000000G  EF   9E  00002              MOVAB   MOM$GB_EVT_PSER, R9                         ;
               58 00000000G  EF   9E  00009              MOVAB   MOM$AB_SERVICE_DATA+8, R8                   ;
               5E      FDE0  CE   9E  00010              MOVAB   -544(SP), SP                                ;
               53        22  AE   9E  00015              MOVAB   EVTBUF+2, PTR                               ; 1061
                             53   DD  00019              PUSHL   PTR                                         ; 1065
       00000000G  00         01   FB  0001B              CALLS   #1, SYS$GETTIM                              ;
                  53         08   C0  00022              ADDL2   #8, PTR                                     ; 1066
               57 00000000G  EF   9A  00025              MOVZBL  MOM$GW_EVT_CODE, R7                         ; 1070
               83            57   B0  0002C              MOVW    R7, (PTR)+                                  ;
               83            03   90  0002F              MOVB    #3, (PTR)+                                  ; 1075
               83 00000000*  EF   90  00032              MOVB    <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_SLI*137>-; 1076
                                                                 >+8>, (PTR)+                                ;
               50 00000000*  EF   9A  00039              MOVZBL  <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_SLI*137>-; 1077
                                                                 >+8>, R0                                    ;
 10         00 00000000*  EF      50   2C  00040              MOVC5   R0, <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_SLI*-; 1081
                                  63       00049                      137>>+9>, #0, #16, (PTR)                    ;
               03            57   91  0004A              CMPB    R7, #3                                      ; 1090
                             09   12  0004D              BNEQ    1$                                          ;
                             83   B4  0004F              CLRW    (PTR)+                                      ; 1092
               83        81  8F   90  00051              MOVB    #-127, (PTR)+                               ; 1094
```

```
                                        83            69  90 00055         MOVB    MOM$GB_EVT_PSER, (PTR)+                      ; 1095
                                        06            57  91 00058 1$:     CMPB    R7, #6                                       ; 1100
                                                      0E  12 0005B         BNEQ    2$                                           ;
                                        83            02  B0 0005D         MOVW    #2, (PTR)+                                   ; 1102
                                        83        81  8F  90 00060         MOVB    #-127, (PTR)+                                ; 1104
                                        83 00000000G  EF  90 00064         MOVB    MOM$GB_EVT_POPR, (PTR)+                      ; 1105
                                        07            57  91 0006B 2$:     CMPB    R7, #7                                       ; 1110
                                                      0E  12 0006E         BNEQ    3$                                           ;
                                        83            03  B0 00070         MOVW    #3, (PTR)+                                   ; 1112
                                        83        81  8F  90 00073         MOVB    #-127, (PTR)+                                ; 1114
                                        83 00000000G  EF  90 00077         MOVB    MOM$GB_EVT_PRSN, (PTR)+                      ; 1115
                                        03            57  91 0007E 3$:     CMPB    R7, #3                                       ; 1120
                                                      05  13 00081         BEQL    4$                                           ;
                                        07            57  91 00083         CMPB    R7, #7                                       ;
                                                      4E  12 00086         BNEQ    10$                                          ;
                                        83  81C30001  8F  D0 00088 4$:     MOVL    #-2117926911, (PTR)+                         ; 1122
                                        51        04  AC  D0 0008F         MOVL    MSGSIZE, R1                                  ; 1127
                                                      06  15 00093         BLEQ    5$                                           ;
                                        63        08  BC  90 00095         MOVB    @MSGADR, (PTR)                               ; 1128
                                                      02  11 00099         BRB     6$                                           ; 1130
                                                      63  94 0009B 5$:     CLRB    (PTR)                                        ;
                                                      53  D6 0009D 6$:     INCL    PTR                                          ; 1128
                                        83        82  8F  90 0009F         MOVB    #-126, (PTR)+                                ; 1132
                                        02            51  D1 000A3         CMPL    R1, #2                                       ; 1133
                                                      0A  15 000A6         BLEQ    7$                                           ;
                                        50        08  AC  D0 000A8         MOVL    MSGADR, R0                                   ; 1134
                                        63        01  A0  B0 000AC         MOVW    1(R0), (PTR)                                 ;
                                                      03  11 000B0         BRB     8$                                           ;
                                        63            01  AE 000B2 7$:     MNEGW   #1, (PTR)                                    ; 1136
                                        53            02  C0 000B5 8$:     ADDL2   #2, PTR                                      ; 1137
                                        83        40  8F  90 000B8         MOVB    #64, (PTR)+                                  ; 1139
                                        04            51  D1 000BC         CMPL    R1, #4                                       ; 1140
                                                      13  15 000BF         BLEQ    9$                                           ;
                                        50        08  AC  D0 000C1         MOVL    MSGADR, R0                                   ; 1143
                                        83        03  A0  90 000C5         MOVB    3(R0), (PTR)+                                ;
                                        51        03  A0  9A 000C9         MOVZBL  3(R0), R1                                    ; 1144
                          63      04    A0            51  28 000CD         MOVC3   R1, 4(R0), (PTR)                             ; 1146
                                                      02  11 000D2         BRB     10$                                          ; 1149
                                                      83  94 000D4 9$:     CLRB    (PTR)+                                       ;
                                        03            57  91 000D6 10$:    CMPB    R7, #3                                       ; 1155
                                                      05  13 000D9         BEQL    11$                                          ;
                                        07            57  91 000DB         CMPB    R7, #7                                       ;
                                                      4C  12 000DE         BNEQ    15$                                          ;
                                                      50  D4 000E0 11$:    CLRL    NODE_CMU                                     ; 1157
                                        52 00000000*  EF  D0 000E2         MOVL    <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_ADD*137>- ; 1158
                                                                                   >+9>, R2                                     ;
                                                      54  D4 000E9         CLRL    R4                                           ;
                                                      52  D5 000EB         TSTL    R2                                           ;
                                                      05  13 000ED         BEQL    12$                                          ;
                                                      54  D6 000EF         INCL    R4                                           ;
                                        50            01  D0 000F1         MOVL    #1, NODE_CMU                                 ; 1159
                                        51 00000000*  EF  9A 000F4 12$:    MOVZBL  <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_NNA*137>- ; 1160
                                                                                   >+8>, R1                                     ;
                                                      55  D4 000FB         CLRL    R5                                           ;
                                                      51  D5 000FD         TSTL    R1                                           ;
                                                      04  15 000FF         BLEQ    13$                                          ;
                                                      55  D6 00101         INCL    R5                                           ;
```

```
                                          50 D6 00103       INCL    NODE_CMU                                    ; 1161
                                          50 D5 00105 13$:  TSTL    NODE_CMU                                    ; 1162
                                          23 15 00107       BLEQ    15$                                         ;
                           83             05 B0 00109       MOVW    #5, (PTR)+                                  ; 1164
              83           50          C0 8F 89 0010C       BISB3   #192, NODE_CMU, (PTR)+                      ; 1166
                           06             54 E9 00111       BLBC    R4, 14$                                     ; 1167
                           83             02 90 00114       MOVB    #2, (PTR)+                                  ; 1169
                           83             52 B0 00117       MOVW    R2, (PTR)+                                  ; 1170
                           0F             55 E9 0011A 14$:  BLBC    R5, 15$                                     ; 1174
                           83          40 8F 90 0011D       MOVB    #64, (PTR)+                                 ; 1178
                           83             51 90 00121       MOVB    R1, (PTR)+                                  ; 1180
              63 00000000* EF             51 28 00124       MOVC3   R1, <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_NNA*-; 1185
                                                                    137>>+9>, (PTR)                             ;
                           03             57 91 0012C 15$:  CMPB    R7, #3                                      ; 1192
                                          56 12 0012F       BNEQ    19$                                         ;
                                          69 95 00131       TSTB    MOM$GB_EVT_PSER                             ; 1194
                                          52 12 00133       BNEQ    19$                                         ;
                           83             07 B0 00135       MOVW    #7, (PTR)+                                  ; 1199
                           56 00000000*   EF D0 00138       MOVL    <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_STY*137>-; 1201
                                                                    >+9>, R6                                    ;
                                          09 12 0013F       BNEQ    16$                                         ; 1203
                           50 00000000G   8F D0 00141       MOVL    #SVD$GK_PCNO_SLO, SVD_INDEX                 ;
                                          1A 11 00148       BRB     18$                                         ;
                           01             56 D1 0014A 16$:  CMPL    R6, #1                                      ; 1204
                                          09 12 0014D       BNEQ    17$                                         ;
                           50 00000000G   8F D0 0014F       MOVL    #SVD$GK_PCNO_TLO, SVD_INDEX                 ;
                                          0C 11 00156       BRB     18$                                         ;
                           02             56 D1 00158 17$:  CMPL    R6, #2                                      ; 1205
                                          07 12 0015B       BNEQ    18$                                         ;
                           50 00000000G   8F D0 0015D       MOVL    #SVD$GK_PCNO_LOA, SVD_INDEX                 ;
                           83          40 8F 90 00164 18$:  MOVB    #64, (PTR)+                                 ; 1208
                           50 00000089    8F C4 00168       MULL2   #137, R0                                    ; 1209
                           83           6840 90 0016F       MOVB    MOM$AB_SERVICE_DATA+8[R0], (PTR)+           ; 1210
                           51           6840 9A 00173       MOVZBL  MOM$AB_SERVICE_DATA+8[R0], R1               ; 1211
              63     01 A840              51 28 00177       MOVC3   R1, MOM$AB_SERVICE_DATA+9[R0], (PTR)        ; 1213
                           83             08 B0 0017D       MOVW    #8, (PTR)+                                  ; 1217
                           83          81 8F 90 00180       MOVB    #-127, (PTR)+                               ; 1219
                           83             56 90 00184       MOVB    R6, (PTR)+                                  ; 1221
                           03             57 91 00187 19$:  CMPB    R7, #3                                      ; 1228
                                          05 13 0018A       BEQL    20$                                         ;
                           07             57 91 0018C       CMPB    R7, #7                                      ;
                                          17 12 0018F       BNEQ    21$                                         ;
              0F 00000000G EF             01 E1 00191 20$:  BBC     #1, MOM$GL_SERVICE_FLAGS, 21$               ; 1230
                           83 06200009    8F D0 00199       MOVL    #102760457, (PTR)+                          ; 1235
              63 00000000* EF             06 28 001A0       MOVC3   #6, <<MOM$AB_SERVICE_DATA+<SVD$GK_PCNO_PHA*-; 1246
                                                                    137>>+9>, (PTR)                             ;
                           50          20 AE 9E 001A8 21$:  MOVAB   EVTBUF, R0                                  ; 1253
          18  AE           53             50 C3 001AC       SUBL3   R0, PTR, EVTDSC                             ;
                     20    AE          18 AE B0 001B1       MOVW    EVTDSC, EVTBUF                              ; 1254
                     1C    AE          20 AE 9E 001B6       MOVAB   EVTBUF, EVTDSC+4                            ; 1255
    10        00           6E             00 2C 001BB       MOVC5   #0, (SP), #0, #16, NFB                      ; 1259
                                       08 AE    001C0                                                           ;
                     08    AE             1C 90 001C2       MOVB    #28, NFB                                    ; 1260
                           6E             10 D0 001C6       MOVL    #16, NFBDSC                                 ; 1261
                     04    AE          08 AE 9E 001C9       MOVAB   NFB, NFBDSC+4                               ; 1262
                           7E             7C    001CE       CLRQ    -(SP)                                       ; 1266
                                       20 AE 9F 001D0       PUSHAB  EVTDSC                                      ;
```

```
                                   0C    AE   9F 001D3          PUSHAB  NFBDSC                                 ;
                  00000000G  EF          04   FB 001D6          CALLS   #4, MOM$NETACP_QIO                     ;
                               00000000'  EF   9F 001DD          PUSHAB  P.AAJ                                  ; 1273
                                   1C    AE   DD 001E3          PUSHL   EVTDSC                                 ; 1272
                                   24    AE   DD 001E6          PUSHL   EVTDSC+4                               ; 1271
                                         07   DD 001E9          PUSHL   #7                                     ; 1270
                  00000000G  EF          04   FB 001EB          CALLS   #4, MOM$DEBUG_MSG                      ;
                                              04 001F2          RET                                            ; 1275

; Routine Size:  499 bytes,    Routine Base:  $CODE$ + 06CF
; 1287          1276  1 %SBTTL 'mom$chk_mop_error  Get volatile byte parameter'
; 1288          1277  1 GLOBAL ROUTINE mom$chk_mop_error (code) : NOVALUE =
; 1289          1278  1
; 1290          1279  1 !++
; 1291          1280  1 ! FUNCTIONAL DESCRIPTION:
; 1292          1281  1 !
; 1293          1282  1 !       Check the status of a MOP function.
; 1294          1283  1 !
; 1295          1284  1 !--
; 1296          1285  1
; 1297          1286  2 BEGIN
; 1298          1287  2
; 1299          1288  2 LOCAL
; 1300          1289  2     msgsize;
; 1301          1290  2
; 1302          1291  2 IF NOT .code THEN
; 1303          1292  3     BEGIN
; 1304          1293  3     mom$bld_reply (mom$ab_msgblock, msgsize);
; 1305          1294  3     $signal_msg (mom$ab_nice_xmit_buf, .msgsize);
; 1306          1295  2     END;
; 1307          1296  2
; 1308          1297  1 END;                                          ! End of mom$chk_mop_error



                                                         0000 00000         .ENTRY  MOM$CHK_MOP_ERROR, Save nothing              ; 1277
                                        5E            04   C2 00002         SUBL2   #4, SP                                       ;
                                        24       04   AC   E8 00005         BLBS    CODE, 1$                                     ; 1291
                                                      5E   DD 00009         PUSHL   SP                                           ; 1293
                                           00000000G  EF   9F 0000B         PUSHAB  MOM$AB_MSGBLOCK                              ;
                             00000000G  EF            02   FB 00011         CALLS   #2, MOM$BLD_REPLY                            ;
                                                      6E   DD 00018         PUSHL   MSGSIZE                                      ; 1294
                                           00000000G  EF   9F 0001A         PUSHAB  MOM$AB_NICE_XMIT_BUF                         ;
                                           02070000   8F   DD 00020         PUSHL   #34013T84                                    ;
                             00000000G  00            03   FB 00026         CALLS   #3, LIB$SIGNAL                               ;
                                                           04 0002D 1$:     RET                                                  ; 1297

; Routine Size:  46 bytes,    Routine Base:  $CODE$ + 08C2
; 1310           1298   1 END
; 1311           1299   1
; 1312           1300   0 ELUDOM



                                                                         .EXTRN  LIB$SIGNAL

;                                        PSECT SUMMARY
;
;        Name                        Bytes                           Attributes
;
;   $CODE$                               2288  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                                136  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                                 Library Statistics
;
;                                              -------- Symbols --------       Pages       Processing
;        File                                  Total      Loaded   Percent     Mapped      Time
;
;   _$255$DUA28:[MOM.OBJ]MOMLIB.L32;1              194          49        25         21      00:00.1
;   _$255$DUA28:[SHRLIB]NMALIBRY.L32;1             887          22         2         47      00:00.2
;   _$255$DUA28:[SHRLIB]EVCDEF.L32;1               213          16         7         15      00:00.1
;   _$255$DUA28:[SHRLIB]NET.L32;1                 1279           3         0         63      00:00.3
;   _$255$DUA28:[SYSLIB]LIB.L32;1                18619          11         0       1000      00:06.5





;                                         COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:MOMSERVIC/OBJ=OBJ$:MOMSERVIC MSRC$:MOMSERVIC/UPDATE=(ENH$:MOMSERVIC)

; Size:          2288 code + 136 data bytes
; Run Time:         00:45.4
; Elapsed Time:     01:26.8
; Lines/CPU Min:      1716
; Lexemes/CPU-Min:  13397
; Memory Used:   287 pages
; Compilation Complete
```

0238 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY